Socio-Affective Computing 2

Basant Agarwal
Namita Mittal *Editors*

Prominent Feature Extraction for Sentiment Analysis

Springer

Socio-Affective Computing

Volume 2

Series Editor
Amir Hussain, University of Stirling, Stirling, UK

Co-Editor
Erik Cambria, Nanyang Technological University, Singapore

This exciting Book Series aims to publish state-of-the-art research on socially intelligent, affective and multimodal human-machine interaction and systems. It will emphasize the role of affect in social interactions and the humanistic side of affective computing by promoting publications at the cross-roads between engineering and human sciences (including biological, social and cultural aspects of human life). Three broad domains of social and affective computing will be covered by the book series: (1) social computing, (2) affective computing, and (3) interplay of the first two domains (for example, augmenting social interaction through affective computing). Examples of the first domain will include but not limited to: all types of social interactions that contribute to the meaning, interest and richness of our daily life, for example, information produced by a group of people used to provide or enhance the functioning of a system. Examples of the second domain will include, but not limited to: computational and psychological models of emotions, bodily manifestations of affect (facial expressions, posture, behavior, physiology), and affective interfaces and applications (dialogue systems, games, learning etc.). This series will publish works of the highest quality that advance the understanding and practical application of social and affective computing techniques. Research monographs, introductory and advanced level textbooks, volume editions and proceedings will be considered.

More information about this series at http://www.springer.com/series/13199

Basant Agarwal • Namita Mittal

Prominent Feature Extraction for Sentiment Analysis

Springer

Basant Agarwal
Malaviya National Institute of Technology
Jaipur, India

Namita Mittal
Malaviya National Institute of Technology
Jaipur, India

Socio-Affective Computing
ISBN 978-3-319-25341-1 ISBN 978-3-319-25343-5 (eBook)
DOI 10.1007/978-3-319-25343-5

Library of Congress Control Number: 2015954185

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

The book is dedicated to my family.

Preface

The field of sentiment analysis is an exciting and new research direction due to a large number of real-world applications. Discovering people's opinion is very important for better decision-making. Sentiment analysis is the study that analyses people's opinion and sentiment towards entities, such as products, services, etc., in the text. It has always been important to know what other people think. People are using online review sites, blogs, forums, social networking sites, etc., for expressing their opinion that increase the user-generated data on the web. Therefore, a necessity of analysing and understanding these online generated data/reviews has arisen. The user can know the merits and demerits of the product from the experiences shared by people on the web, which can be useful for them in decision-making. E-commerce companies can improve their product or services on the basis of people's opinion and current trends. The automatic analysis of online contents to extract opinion requires deep understanding of natural text by the machine, but capabilities of most of the existing models are known to be unsatisfactory.

Two types of approaches have been used in the literature for sentiment analysis: (i) machine learning approach and (ii) semantic orientation approach. Machine learning approaches face main challenges as (i) machine learning approaches produce high-dimensional feature vector consisting of noisy, irrelevant and redundant features. Most of the existing feature selection techniques, used for sentiment analysis, do not consider the redundancy among the features. Existing methods select the important features based on goodness criteria for the class attribute. (ii) Generally, generated feature vector has to deal with problem of data sparsity.

Semantic orientation approaches are categorized into corpus-based and lexicon-based (knowledge-based) approaches. Corpus-based approaches mainly depend on the method to determine the polarity of the words. These approaches do not perform well because polarity of words changes with the domain and context, and there is no such corpus available which can provide polarity of words depending on the domain and context. Knowledge-based approaches depend on the already developed knowledge bases like SentiWordNet, WordNet, etc. The problem with these approaches is the coverage of knowledge bases as most of the available knowledge bases contain general knowledge (not affective knowledge) which is insufficient to determine the polarity of the document.

The objective of this book is to improve the performance of the sentiment analysis model by incorporating the semantic, syntactic and common-sense knowledge. This book presents the semantic concept extraction method that uses dependency relations between words to extract the features from the text. Proposed approach combines the semantic and common-sense knowledge for the better understanding of the text. In addition, the book also presents novel methods to extract prominent features from the unstructured text by eliminating the noisy, irrelevant and redundant features. This book also aims to propose a method for efficient dimensionality reduction to alleviate the data sparseness problem being faced by machine learning model. The main findings of this book are as follows.

1. Performance of the sentiment analysis can be improved by reducing the redundancy among the features. In this book, experimental results show that minimum Redundancy-Maximum Relevance (mRMR) feature selection technique improves the performance of the sentiment analysis by eliminating the redundant features.
2. Boolean Multinomial Naive Bayes (BMNB) machine learning algorithm with mRMR feature selection technique performs better than Support Vector Machine (SVM) classifier for sentiment analysis.
3. The problem of data sparseness is alleviated by semantic clustering of features, which in turn improves the performance of the sentiment analysis.
4. Semantic relations among the words in the text have useful cues for sentiment analysis. Common-sense knowledge in form of ConceptNet ontology acquires knowledge, which provides a better understanding of the text that improves the performance of the sentiment analysis.
5. Considering the importance of the feature with respect to the domain improves the performance of the sentiment analysis.
6. Splitting of the multi-word features improves the performance of sentiment analysis for the domains having only limited labelled dataset.

All the experiments are performed on four standard datasets, viz. movie review dataset provided by Cornell University and product review dataset (i.e. book, DVD, electronics) consisting of Amazon reviews. Experimental results show the effectiveness of all the proposed methods over state-of-the-art methods.

Jaipur, India Basant Agarwal
December, 2015

Acknowledgements

This book would not be possible without the contribution of many individuals, to whom I express my appreciation and gratitude. Mostly the contents of this book are taken from my research work done for PhD. First of all, I am deeply indebted to my supervisor **Dr. Namita Mittal** for the knowledge and experience she shared with me. She was always available whenever I needed her guidance. Her invaluable suggestion and advice helped me a lot in nurturing my research work in the right direction. She is very kind and supportive. It is impossible for me to imagine having a better supervisor for my PhD. It was really an honour to work with her.

I would especially like to thank **Dr. Erik Cambria** for permitting me to be an intern at Temasek Laboratories at the National University of Singapore and for all the knowledge he and his group shared with me during my internship period. I would like to thank Soujanya Poria with whom I met there; discussions with them on various common-sense and SenticNet-based sentiment analyses were very useful.

I am grateful to **Dr M. C. Govil**, Head of Department, Computer Engineering, Malaviya National Institute of Technology, Jaipur, for his kind cooperation. I am also grateful to all the members of my doctoral guidance committee: **Dr. Neeta Nain**, **Dr. Girdhari Singh**, **Dr. Dinesh Goplani** and **Dr. D. Boolchandani**; their constructive suggestions helped me a lot to improve my PhD work. I would like to thank all the learned faculty members, especially **Dr. Vijay Laxmi** and **Prof. M.S. Gaur**, for their support and help during this period. I am also thankful to the non-teaching staff of the Department of Computer Engineering for all the help.

My special gratitude also goes to all the persons from whom I have received critical reviews during this period: **Prof. Alexandra Gelbukh** (Research Professor, Center for Computing Research, CIC, National Polytechnic Institute, IPN, Mexico), **Prof. Pushpak Bhattacharya** (Professor, Indian Institute of Technology, Bombay, India), **Prof. R. K. Agarwal** (Professor, Jawaharlal Nehru University, Delhi, India) and **Prof. Niladri Chatterjee** (Professor, Indian Institute of Technology, Delhi, India) who visited MNIT during this period. Their valuable suggestions and feedback were useful for my PhD work.

I would like to thank MNIT Jaipur for providing me the PhD scholarship during this period.

My special thanks to all my wonderful friends and fellow researchers who made the whole environment of the lab very refreshing. They celebrated my achievements and cheered me up in my difficult moments. Discussion with co-researchers helped me to remove the small mistakes in the research work. I would like to thank my co-researchers Mr. Lokesh Kumar Sharma and Mr. Ankit Vidhyarthi, who helped me in proofreading the book.

And last but not least, my sincere thanks go to my family and friends for their endless encouragement and support during the course of my studies. Their love and care always motivated me to work harder and harder. For any inadequacies that may remain in this work, the responsibility is entirely my own.

Signed: Basant Agarwal

Date: December 1, 2015

Contents

Acronyms

AI	Artificial intelligence
ANN	Artificial neural network
BMNB	Boolean Multinomial Naive Bayes
BoW	Bag-of-words
CHI	Chi-square
CSR	Class sequential rules
DF	Document frequency
EWGA	Entropy Weighted Genetic Algorithm
FN	False negative
FP	False positive
GI	General Inquirer, a polarity lexicon
IG	Information gain
IMDb	Internet Movie Database
IT	Information technology
KNN	K-nearest neighbor
LSA	Latent semantic analysis
ME	Maximum entropy
MI	Mutual information
ML	Machine learning
MLP	Multilayer perceptron
mRMR	Minimum Redundancy Maximum Relevance (mRMR)
NB	Naive Bayes
NLP	Natural language processing
NN	Neural network
PCA	Principal component analysis
PMI	Point-wise mutual information
POS	Part of speech
RF	Random forest
SVD	Singular value decomposition
SO	Semantic orientation
SVM	Support vector machine

SWN	SentiWordNet, a lexicon containing the polarity values of words
TF	Term frequency
TF-IDF	Term frequency-inverse document frequency
TP	True positives
VSM	Vector space model
WNA	WordNet-Affect

List of Figures

List of Tables

Chapter 1
Introduction

The textual information available on the Web is of two types: facts and opinions statements. Facts are objective sentences about the entities and do not show any sentiments. Opinions are subjective in nature and generally describe the people's sentiments towards entities and events. Most of the existing research with the available online text has been emphasized on the factual data in various natural language processing (NLP) tasks, e.g., inform retrieval [69], text classification [41], etc. Research on processing the opinionated sentences is still very limited due to a large number of challenges involved in the field [20, 67].

Sentiment analysis research has been increasing tremendously for last 10 years due to the wide range of business and social applications [19, 67, 91]. Opinion mining or sentiment analysis is the study that analyzes people's opinion and sentiment towards entities such as products, services, etc., in the text. The automatic analysis of online contents to extract the opinion requires deep understating of natural text by the machine. Sentiment analysis research can be categorized among document-level [91], sentence-level [73], and aspect-/feature-level sentiment analysis [52]. Document-level sentiment analysis classifies a review document as containing positive or negative polarity. It considers a document as a single unit. Sentence-level sentiment analysis takes a sentence to extract the opinion or sentiment expressed in that sentence. Both the document-level and the sentence-level sentiment analyses do not detect what exactly people liked and did not like. Aspect-based sentiment analysis deals with the methods that identify the aspects/entities in the text about which an opinion is expressed [67]. Further, the sentiments expressed about these entities are identified. For example, "although the service is not that good, I still love the food"; in this example, "service" and "food" are two entities about which opinion is expressed. Aspect-based sentiment analysis model, firstly, identifies these entities, and further, opinions about these entities are determined. Aspect-based sentiment analysis is also called feature-level opinion mining [52, 67]. The development of techniques for the document-level sentiment analysis is one of the significant components of this area. A lot of research is available in literature for

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_1

detecting sentiment from the text [1, 21, 67, 93, 103, 117, 130]. Still, there is a huge scope of improvement of these existing sentiment analysis models. In this book, new methods proposed to extract features from the unstructured text that can include syntactic, semantic, and common sense knowledge.

Techniques employed by sentiment analysis models can be broadly categorized into machine learning [93] and semantic orientation approaches [29, 132]. Machine learning model requires large training dataset, whereas semantic orientation approach detects the polarity of the words on the basis of the corpus or dictionary. The process to construct the machine learning model for sentiment analysis is as follows. Initially, intelligent features are extracted from the text that can incorporate the semantic, syntactic, sentiment, and commonsense knowledge because simple keywords may not convey accurate sentiments of the user. Next, an appropriate weighting scheme is required to give weight to each feature according to their importance. Further, an efficient feature selection technique is required to select only important features by eliminating the noisy and irrelevant features for better classification results. Finally, a robust machine learning method is required for the classification.

Semantic orientation-based approach for sentiment analysis works in three steps. Initially, sentiment-rich features are extracted from the unstructured text. Sentiment-rich feature means the feature that contains direct opinion of the user; for example, "good book" expresses a positive opinion. Further, semantic orientations of these features are determined based on two methods, i.e., corpus based or dictionary based. Finally, the overall polarity of the document is computed by aggregating the semantic orientations of all the features. Semantic orientation-based approaches are classified into two types based on the methods used to compute the polarity of the features: (i) corpus based and (ii) dictionary or lexicon or knowledge based [29]. In corpus-based approach, polarity value is computed based on the co-occurrences of the term with other positive or negative seed words in the corpus. The main motivation behind this approach is that the semantic orientation of any feature is said to be positive if it has association with positive seed words (e.g., excellent). Similarly, it is said to be negative semantic orientation if it has association with negative seed words (e.g., bad, poor, etc). Whereas, dictionary-based approaches determine the polarity of a feature based by utilizing the pre-developed polarity lexicons like SentiWordNet [37], WordNet [75], SenticNet [19], etc. These methods are also called as lexicon-based or knowledge-based approaches.

In this book, we propose a novel approach for sentiment analysis that incorporates semantic and commonsense knowledge. The proposed concept extraction approach exploits the relationship between words; it obtains the semantic relationship between words based on dependency parsing. We investigate the importance of commonsense knowledge obtained from ConceptNet for the sentiment analysis model. We explore various feature extraction and selection techniques to mine the prominent features for machine learning model.

1.1 Motivation

What other people think has always been very important in decision making. Whenever people want to purchase a product, e.g., mobile phone, camera, laptop, etc., they ask their friends or their peers about the product if they have used that. Nowadays, due to the advent of Web recent trends, people express their opinions, feelings, and experiences about the products or services on the forums, blogs, social network, and content-sharing services. The user can know the merits and demerits of the product from the experiences shared by people on the Web, which can be useful for them in taking purchasing decisions [9, 10]. E-commerce companies can improve their products or services on the basis of users' opinion and can also know the current trends of the market. Examples of sentiment analysis include identifying movie popularity from online reviews, which model of a camera is liked by most of the users, which music is liked by most of the people, etc. Opinion mining and sentiment analysis also have applications in political domain and brand analysis. Government may develop sentiment analysis model which can accumulate the common people's opinion about the government policies. These models can help the government to improve and further develop the policies for the welfare of the people. Nowadays, large numbers of companies are investing money in developing marketing strategies and knowing the opinion of the people for their products and brands. In addition, several companies provide the sentiment analysis tools to such companies to know the trend of their products in the public.

Machine learning approaches perform very well if a large labeled dataset is available for the training. Machine learning-based sentiment analysis model faces the following problems. Firstly, machine learning-based approaches represent the text by constructing feature vector with bag-of-words [10, 13, 93]. Generally, the size of this generated feature vector is of tens of thousands, comprising of lots of noisy and irrelevant features. Secondly, feature vector generated for classification is very sparse; therefore, it limits the performance of the machine learning algorithms. Thirdly, machine learning algorithms are sometimes biased towards negative reviews. The possible reason for this observation is that people generally use more positive words as compared to negative words to express their sentiments. Therefore, positive words occur in both the negative and positive classes, hence very less discriminating. In contrast, negative words are more discriminating, making machine learning model to work better for negative reviews.

Semantic orientation-based methods may perform in a biased way for positive reviews. The possible reason is that semantic orientations of positive opinionated words are computed more than negative words; therefore, for a testing document, it is highly probable that semantic orientation of positive words would be available more from the training corpus in contrast to negative words. Knowledge-based approaches depend on the already developed general knowledge bases like SentiWordNet, WordNet, etc. The main problem with these approaches is the coverage as it is critical to have such a large knowledge base which can provide polarity information for every domain. In addition, most of the available knowledge bases contain general knowledge (not affective sufficient knowledge) which is insufficient to determine the polarity of the documents. Corpus-based approaches mainly rely on the method which is used to determine the polarity of words. These approaches do not perform well because the polarity of words changes with the domain and context, and there is no such accurate polarity computation method available which can provide polarity of words depending on the domain and context. In addition, corpus-based methods require large training corpus to compute the polarity of words. Knowledge-based methods perform well only if the knowledge base is very huge and provides accurate polarity information. Knowledge-based approaches have a big advantage of not requiring large training corpus. It just requires one-time effort of developing a large knowledge base. However, machine learning algorithms may learn the patterns found in the labeled dataset and can accurately determine the polarity of the testing documents. Machine learning-based approaches can learn domain-specific patterns from the text, resulting into better classification results as compared to semantic orientation-based approaches.

The problems with the existing sentiment analysis models are as follows:

1. In the literature, researchers have proposed various methods for sentiment analysis that are mostly based on bag-of-words, syntactic, and semantic information. As most of the existing methods do not consider all these information all together, the performance of the sentiment analysis can be improved. In this book, new methods are introduced which include syntactic, semantic, and commonsense knowledge all together for sentiment analysis.
2. Most of the existing feature selection techniques used for sentiment analysis does not consider the redundancy among the features. Existing methods select the important features based on goodness criteria for the class attribute. For example, information gain, chi-square, mutual information, etc., are popular feature selection techniques for sentiment analysis, which select the important features based on the relevance of feature with the class attribute; they do not consider redundancy among features. In this book, new feature selection method is proposed which focuses on selecting the features for sentiment analysis considering both relevancy to the class attribute and also redundancy among the features.
3. Most of the existing machine learning-based sentiment analysis model focus either on data sparseness or semantics independently for improving the performance. Data sparseness as well as semantics have not been considered together.
4. As most of the existing sentiment analysis models do not consider commonsense knowledge with machine learning for sentiment analysis, which is intuitively very useful for sentiment analysis.
5. Most of the existing sentiment analysis models don't consider the importance of the product feature in determining the overall sentiment of the document. For example, "the audio quality of this phone is awesome, but the pictures taken by its camera are not good." In most of the existing sentiment analysis models, the above sentence may produce neutral or negative sentiment. But, for most of the people, "audio quality" of the phone is more important than "picture quality" of the phone.

Chapter 2
Literature Survey

Sentiment analysis research has attracted a large number of researchers around the globe [61, 66, 93, 127]. Sentiment analysis attempts to determine whether a given text is subjective or objective and further, whether a subjective text contains positive or negative opinion. Techniques employed by sentiment analysis models can be broadly categorized into machine learning [93] and semantic orientation approaches [29, 132]. A lot of research has been done for detecting sentiment from the text [67]. Still, there is a huge scope of improvement of these existing sentiment analysis models. The performance of the existing methods can be further improved by including more semantic information.

Sentiment analysis techniques are categorized into machine learning (ML) and semantic orientation (SO) approaches. In this chapter, we discuss the research work initiated by various researchers in the domain of document-level sentiment classification.

2.1 Machine Learning Approaches

Sentiment analysis using machine learning is generally treated as a text classification problem. Traditional topic-based text classification problem is easier than sentiment classification. Topic-related keywords are the key features for classification of documents into different categories like "politics," "sport," etc. However, in case of sentiment detection, sentiment-bearing words like "good," "great," "amazing," etc. are important [46, 93]. Machine learning methods have been widely applied mostly for document-level sentiment analysis [93]. Machine Learning-based approaches for sentiment analysis work in following phases:

1. Text preprocessing
2. Feature selection methods

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*, Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_2

3. Feature weighting and representation schemes
4. Machine learning algorithm

In literature, researchers have proposed techniques in each phase to improve the performance of overall sentiment analysis. Research done in each of these phases is discussed in subsequent sections.

2.1.1 Text Preprocessing

Supervised machine learning methods have been widely applied for sentiment analysis [1, 40, 93, 124, 144]. Pang et al. [91] are the first to use unigrams, bigrams, position-based features, POS-based features, adjectives, adverbs, and their combination as features for supervised document-level sentiment analysis. Their experimental results showed that unigram features performs best among these features. Dave et al. [30] also applied supervised learning approach for sentiment analysis of multiple product reviews with more sophisticated features based on linguistic knowledge with feature selection and smoothing techniques. Their experimental results showed that bigram and trigram can improve the performance of sentiment analysis under certain settings. Ng et al. [81] extracted features from the text that contain syntactic and semantic information and used them to build machine learning model for sentiment analysis.

Matsumoto et al. [72] has used syntactic relation between words and words subsequence as features for sentiment analysis. Further, they used frequent mining algorithm to determine the important features for the sentiment analysis on movie review dataset. Pak and Paroubek [89] proposed to use subgraphs from the dependency tree of a parsed sentence as features for sentiment analysis. They construct the feature vector based on the extracted subgraphs and further develop machine learning model for sentiment analysis. Their experimental results showed that subgraphs-based features with support vector machine (SVM) classifier outperforms other bag-of-words and n-gram features on movie review dataset. This standard dataset is known as Cornell Movie Review Dataset; it consists of 2000 reviews that contain 1000 positive and 1000 negative reviews collected from Internet Movie Database (IMDb). It can be obtained from this URL (http://www.cs.cornell.edu/people/pabo/movie-review-data/). Nakagawa et al. [80] have used syntactic dependency trees for sentiment analysis and obtained better performance than bag-of-words features. Xia and Zong [143] extracted features that include the information of relation between words as features for sentiment analysis. Their results showed the effectiveness of the proposed method. Gamon [42] investigated that deep linguistic features derived from phrase structure trees and part of speech annotations in addition to word n-gram features improve the performance of sentiment analysis. They performed linguistic analysis of the data part-of-speech information coupled with semantic relations (e.g., “verb-subject-noun” indicating a nominal subject to a verbal predicate). Riloff et al. [109] used subsumption hierarchy to formally define various types of lexical features. Joshi and Penstein-Rose [54] also experimented with syntactic dependency relation-based features for sentiment analysis, and further, they proposed methods to generalize dependency features for sentiment analysis model. Tu et al. [131] proposed a feature extraction approach that extracts features from the dependency forest. Dependency forest is a compact representation of multiple dependency trees. Their experimental results showed improvement in the performance of sentiment analysis on movie review dataset.

Mejova and Srinivasan [74] experimented the effectiveness of various Part-of-Speech (POS)-tagged features for supervised sentiment analysis. They used adjectives, adverbs, and nouns in combination and also as independent features. Their experimental results showed that adjectives, adverbs, and nouns in combination perform better than as independent features. It has also been seen that adjective features outperform others as individual POS-tagged feature. Mullen and Collier [79] proposed a method to expand the feature set based on Osgood's theory of semantic orientation [88] and Turney's semantic orientation [132] for supervised sentiment analysis. Nguyen et al. [83] proposed new rating-based features for document-level sentiment analysis. Further, authors presented the results by combining the rating-based features with unigrams, bigrams, and trigrams on the movie review dataset.

Dang et al. [29] proposed a lexicon-enhanced method by combining machine learning and semantic orientation-based approaches that significantly improve the performance of sentiment analysis. They proposed sentiment features based on SentiWordNet lexicon [37] in addition to content free, content-specific features like unigram, bigrams. Their experimental results showed that rarely used sentiment features enhance the performance of sentiment analysis. Whitelaw et al. [139] proposed appraisal group-based features for enhancing the performance of sentiment analysis. Appraisal groups are phrases that contain word groups like "very beautiful," "extremely bad," etc., unlike individual words. These word groups are intensifiers or modifiers (i.e., very, extremely, etc.) to opinion words (i.e., beautiful, bad, etc.). These appraisal groups are used as features with bag-of-words features for sentiment analysis, resulting into improved performance.

Text needs to be preprocessed as tokens or strings before applying machine learning models. In preprocessing, stemming (reducing words to their stem or root form) and stop word removal methods are generally used for sentiment analysis [91]. Negation word (no, not, never, didn't, don't, can't) reverses the polarity of the sentence. Therefore, it is important to handle negation for sentiment analysis. Pang et al. [91] have added the tag NOT to every word between a negation word ("no", "isn't", "didn't", etc.) and the first punctuation mark following the negation word. For example, "this is not a good movie," polarity of the word "good" is reversed by "not." After negation handling, the sentence becomes " this is *NOT_a NOT_good NOT_movie*."

Feature extraction from unstructured text is a key step in developing sentiment analysis models. It is noted that most of the existing methods develop machine learning models based on unigrams, bigrams, trigrams, and dependency features. Some researchers used Part-of-Speech (POS) information to include linguistic knowledge in developing machine learning-based sentiment analysis model. Further, researchers used syntactic information in machine learning models with the help of dependency relations. Syntactic information proved to be useful for sentiment analysis. Overall, researchers have proposed various methods for sentiment analysis that are mostly based on bag-of-words and syntactic and semantic information. As most of the existing methods do not consider all these information all together, the performance of the sentiment analysis can be improved. In this book, due to the importance of the feature extraction process in the performance of the sentiment analysis, various new composite features are constructed using unigrams, bigrams, POS-based features, and dependency features. We also proposed a new semantic parsing scheme which extracts semantic features from the unstructured text. Proposed method to construct feature vector also include the commonsense knowledge in building machine learning model for sentiment analysis. In addition, various new feature extraction methods are proposed in this book that outperforms existing methods for sentiment analysis. Machine learning models developed with multi-word features face the problem of data sparseness. To alleviate the data sparseness, a new feature extraction method is proposed in this book, namely, clustering features which constructs the feature vector by grouping the similar semantic features.

2.1.2 Feature Selection Methods

A feature is relevant for the classification if, by eliminating this feature, the performance degrades. Irrelevant features are not necessary at all for the classification [146]. Redundant features degrade the performance of the classification, and redundancy among the features can be detected by the measure of correlation among the features [94]. In the presence of thousands of features, it is very common that a large number of features are not informative due to irrelevancy or redundancy with respect to the class. Therefore, removal of these irrelevant and redundant features can improve the performance of the classification [7, 41, 146]. Characteristic of salient features for sentiment analysis is that it should be discriminating enough so that classifier can use the information of presence of this feature to predict the class for new sample [50]. For example, if a term “excellent” occurs predominantly in positive class very frequently, then the presence of this term in a new test document indicates that the document belongs to the positive class.

Apart from using efficient feature extraction technique and assigning correct weights, feature selection methods are also very important for the performance of the machine learning models. The main purpose of feature selection methods is to select the features which are relevant and discriminating for the classification by eliminating the noisy and irrelevant features. Performance of the machine learning methods deteriorates due to high feature vector length and inclusion of irrelevant and noisy features [16]. Dimension of feature vector can be reduced by using feature selection method, i.e., information gain (IG) [69], mutual information (MI) [69], etc., or feature transformation method, i.e., singular value decomposition (SVD), etc. Feature selection methods select the important features using some goodness of a term formula; top features are selected above the threshold criteria, and other irrelevant features are dropped. Feature transformation methods convert the high-dimensional feature vector into lower-dimensional feature space, and reduced feature vector contains the contribution of each feature of initial feature vector. Wang and Wan [136] applied latent semantic analysis (LSA) method for dimension reduction; further reduced feature vector is used by support vector machine (SVM) for improving the performance of sentiment analysis. In literature, mostly feature selection techniques are used to reduce the feature vector length because it is easy to use and computationally efficient as compared to other feature transformation methods.

A lot of feature selection methods have been proposed by researchers in the literature for the sentiment analysis like IG, MI, chi-square (CHI), document frequency (DF), etc. [3, 5, 93, 124]. The simplest feature selection technique is the document frequency (DF). DF feature selection method uses the most frequently occurring terms in the corpus to construct feature vector. This approach is commonly used with general text classification as well as sentiment analysis [91]. Tan and Zhang [124] experimented with four feature selection methods (MI, IG, CHI, and DF) for sentiment analysis on Chinese documents and with five machine learning algorithms, i.e., K-nearest neighbor, centroid classifier, winnow classifier, naive Bayes (NB), and support vector machine (SVM). They observed that IG performs best among all the feature selection methods and SVM gives best results among machine learning algorithms. Abbasi et al. [1] examined that IG and genetic algorithm improves the accuracy of sentiment analysis for movie review dataset. They also proposed hybrid approach entropy weighted genetic algorithm (EWGA) by combining the IG and genetic algorithm. Nicholls and Song [84] proposed a new feature selection method, namely, document frequency difference, and further compared it with other feature selection methods for sentiment analysis. Authors in [42] used log-likelihood method to select important features for sentiment analysis.

Agarwal and Mittal [4] proposed Categorical Probability Proportion Difference (CPPD) feature selection method, which is capable of selecting the features which are relevant and capable of discriminating the class. Categorical Probability Proportion Difference (CPPD)-based feature selection method is a combination of the Categorical Proportional Different (CPD) and Probability Proportion Difference (PPD) methods. Categorical Proportional Different (CPD) method measures the degree to which a term contributes in discriminating the class, and top contributing terms are selected for classification [116]. Whereas, PPD method measures the degree of belongingness or probability that a term belongs to a particular class, and term with higher degree of belongingness are considered for the classification. The benefit of CPD method is that it measures the degree of class-distinguishing property of a term, which is an important attribute of a prominent feature. It can eliminate terms, which are occurring in both the classes equally and are not important for the classification. It can easily eliminate the terms with high document frequency but are not important for classification like stop words (i.e., a, the, an etc.). However, PPD value of term indicates the belongingness/relatedness of a term to the classes and difference measures the class discriminating ability. It can remove the terms with less document frequency, which is not important for sentiment analysis like rare terms. Wang et al. [135] proposed a new Fisher's discriminant ratio-based feature selection method for text sentiment analysis.

Duric and Song [35] proposed a new feature selection method that uses content and syntax model to separate the entities under review and the opinion context (i.e., sentiment modifiers). Their experimental results showed that using these features with maximum entropy classifier provides competitive results with the state-of-art approaches. Agarwal and Mittal [5] investigated the hybrid feature selection method combining IG and rough sets for sentiment analysis. Abbasi [2] proposed an intelligent feature selection method that can exploit the syntactic and semantic information from the text. The proposed approach illustrates that a heterogeneous feature set coupled with appropriate feature selection method can improve the performance of sentiment analysis. O'keefe and Koprinska [86] introduced two new feature selection methods for sentiment analysis, i.e., SentiWordNet Subjectivity Score (SWNSS) and SentiWordNet Proportional Difference (SWNPD). The SWNSS method is able to distinguish objective and subjective terms, since only subjective terms should carry sentiment. SWNPD is able to incorporate the class discriminating ability for feature selection. Verma and Bhattacharyya [134] initially pruned the semantically less important terms based on semantic score retrieved from SentiWordNet [37]; further IG feature selection technique is used to select important features for improved classification accuracy.

Various feature selection methods have been proposed in the literature for sentiment analysis, viz., IG, MI, CHI, DF, etc. Feature selection methods have been proved to be very useful for the sentiment analysis. Feature selection methods eliminate the noisy and redundant information from the feature vector that in turn improve the performance of the sentiment analysis with respect to accuracy and execution time. The main focus of existing feature selection models is to select relevant features for classification. They do not consider redundancy can be reduced among the features. In this book, mRMR feature selection method is used to select relevant features with minimizing the redundancy among the features.

2.1.3 Feature Weighting and Representation Schemes

In machine learning-based approaches, feature weighting schemes are very important. It assigns weights to the features according to their sentiment importance for improved classification results. In sentiment analysis, many weighting schemes have been proposed in literature such as term frequency (TF), binary weighting scheme, term-frequency-inverse document frequency (TF-IDF), etc. [93]. In binary weighting scheme, feature value is 1 if a term is present; otherwise, it is 0. In TF-IDF weighting method, weights are given to each term according to how rare these terms are in other documents. Its weights are computed by $w_{ij} = tf_{ij} * idf_i$, where tf_{ij} is the frequency of term i in document j, and idf_i is the inverse document frequency which is equal to $\log(N/n_i)$, N is total number of documents in the corpus, and n_i is number of documents containing the term i [69].

In most of the topic-based text classification, TF-IDF weighting scheme performs well, but in sentiment analysis, binary weighting scheme performs better as compared to other frequency-based schemes [91]. One possible reason for this observation is that people tend to use different sentiment words to express their sentiments in writing reviews. For example, a person writes a review about a camera like this, "This Nikon camera is great. Picture quality is clear and it looks nice." Here, the author has used different sentiment words like "great," "clear," and "nice." It is very unlikely that he would write the same review with only one sentiment word "great" as "this Nikon camera is great. Picture quality is great and its looking is also great." In this example, it is clear that presence/absence of term is more important than frequency of that term for sentiment analysis.

Deng et al. [32] proposed a weighting scheme based on the importance of a term in the document and importance of the term in expressing sentiment for supervised sentiment analysis. Martineau and Finin [71] proposed a new feature weighting scheme, namely, Delta TF-IDF, which performed better than term frequency and TF-IDF weighting scheme for sentiment analysis. In Delta TF-IDF, more weights are given to the terms which are unevenly distributed among positive and negative classes. It measures the importance of a feature based on the class distinguishing property of a term. The value of an evenly distributed feature is zero. The more uneven the distribution of the feature among classes, the more important a feature should be.

Dai et al. [28] highlighted the sentiment feature by increasing their weights for improving the performance of the sentiment analysis. Further, bagging is used to construct multiple classifiers on various feature spaces and finally combined them into an aggregate classifier for improved performance. Paltoglou and Thelwallm [90] presented a detailed study of various weighting schemes for sentiment analysis. Their experimental results showed that variants of classic tf-idf scheme can improve the performance of sentiment analysis. Authors in [63] proposed an improved feature weighting method, which assigns the weights to the terms by sentiment scores by mutual information. Their experimental result showed that mutual information-based weighting method is more effective than others. There has been work in incorporating positional information of the words in text. In this direction, Raychev and Nakov [107] proposed a new language-independent weighting scheme based on the word position and its likelihood of being subjective. Next, they used multinomial naive Bayes with the position information in the feature set. The idea behind their work was that the use of a particular word can have different subjective power depending on where it occurs in the document. Authors conducted their experiments on the movie dataset.

O'Keefe and Koprinska [86] experimented with various feature weighting methods like feature frequency, feature presence (FP), and TFIDF. Further, authors proposed three other methods based on words grouped by their SentiWordNet

(SWN) values, i.e., SWN Word Score Group (SWN-SG), SWN Word Polarity Groups (SWN-PG), and SWN Word Polarity Sums (SWN-PS).

Various weighting schemes have been proposed in literature such as TF, binary weighting scheme, TF-IDF, etc. Researchers have experimented various weighting schemes with the idea of using polarity information in assigning the weights to the features. Among them binary weighting scheme has been considered best for sentiment analysis.

2.1.4 Machine Learning Algorithms

Machine learning methods have been widely applied for sentiment analysis problem [7, 8, 25, 26, 91, 99]. Reduced and optimal feature vector, generated after the feature selection method, is used by the machine learning model. Mainly support vector machine (SVM), naive Bayes (NB), maximum entropy [142], and artificial neural networks methods have been adopted by most of the researchers for sentiment analysis [76, 91, 93, 124, 149]. Pang et al. [91] used different machine learning algorithms like NB, SVM, and maximum entropy (MaxEnt) for sentiment analysis of movie review dataset. Their experimental results showed that SVM outperforms other machine learning method for sentiment analysis. Tan and Zhang [124] also explored that SVM performs better than other classifiers for sentiment analysis. O'Keefe and Koprinska [86] used SVM and NB classifiers for sentiment analysis with various feature weighting schemes and feature selection methods. Their experimental results showed that SVM classifier is better than NB classifier for sentiment analysis. Ye et al. [145] experimented with three supervised learning algorithms, namely, NB, SVM, and character-based N-gram model for the domain of travel destination reviews. They used frequency of words to give weights to the features. Their experimental results showed that SVM outperforms other classifiers for sentiment analysis. Cui et al. [27] also investigated that discriminative classifiers such as SVM is more appropriate for sentiment analysis as compared to other generative models and winnow classifier.

Moraes et al. [76] presented an empirical study on the comparison between SVM and artificial neural network (ANN) classifiers for document-level sentiment analysis. They discussed the limitations of both the methods for sentiment analysis. Saleh et al. [111] experimented SVM classifier with several weighting schemes for various domains of datasets. Li et al. [62] constructed various classifiers with different feature sets of unigrams and POS-based features, and further, these classifiers are combined using several combination rules. It was investigated that combined classifier outperforms individual classifiers. Tsutsumi et al. [129] also proposed an integrated classifier consisting of SVM, maximum entropy, and score calculation-based classifiers, resulting into improved performance of movie review classification. Osajima et al. [87] have proposed a method for polarity classification of sentences in review documents based on aggregation of word polarity. Authors in [64] developed a three-phase framework for choosing optimal combination of classifiers based on assembling multiple classifiers. Xia et al. [144] proposed various types of ensemble methods for various categories of features (i.e., POS based, word relation based) and classifiers (NB, SVM, maximum entropy) for sentiment analysis. Prabowo and Thelwall [104] proposed hybrid classifier by combining rule-based classification, machine learning, and supervised learning method to improve the classification effectiveness. Dinu and Iuga [34] studied the use of naive Bayes classifier for opinion mining applications. Kang et al. [57] proposed a method to improve the NB and SVM classifiers for restaurant reviews.

The field of machine learning has provided many models that are used to solve sentiment analysis problem. Naive Bayes, SVM, decision trees, maximum entropy, and hidden Markov models are most popular among them for sentiment analysis. However, SVM has been considered as best machine learning algorithm for sentiment analysis. However, Agarwal and Mittal [10] showed that BMNB classifier with mRMR feature selection method can perform better than SVM classifier for sentiment analysis. They showed that the performance of the machine learning models can be improved by including more informative and less redundant features.

2.2 Semantic Orientation-Based Approaches

A word is said to have positive semantic orientation if it is used to convey favorable sentiment. For example, positive semantic orientation words are “excellent,” “happy,” “honest,” etc. Similarly, if a word is conveying unfavorable sentiment, then it is said to have negative semantic orientation, for examples, “poor,” “sad,” “dishonest,” etc. To determine the semantic orientation of words is crucial for sentiment analysis.

Two types of techniques have been described in literature for semantic orientation-based approach for sentiment analysis, viz., corpus based and dictionary based [14, 15, 29]. In corpus-based approach, polarity value is computed based on the co-occurrences of the term with other positive or negative seed words in the corpus; there are various methods reported in the literature to determine the polarity value of the term. Whereas, dictionary-based approaches utilize the pre-developed polarity lexicons like SentiWordNet [37], WordNet [75], SenticNet [19], etc. These methods are also called as lexicon-based or knowledge-based approaches. Semantic orientation-based approaches for sentiment analysis work in following phases. Initially, sentiment-rich features are extracted from the unstructured text. Further, semantic orientations of these sentiment-rich features are determined based on corpus or dictionary. Finally, overall polarity of the document is determined by aggregating the semantic orientations of all the features [132].

Initial work for identifying the semantic orientation of words is done by Hatzivassiloglou and McKeown [47]. They developed a supervised learning method to compute the semantic orientation of adjectives. Esuli and Sebastiani [36] proposed a method to determine the semantic orientation of subjective words based on quantitative analysis of the glosses of these words. Turney [132] proposed an unsupervised method for identifying the polarity of the movie review documents. They extracted two-word phrases using fixed POS-based patterns, where one of the words in the phrase was an adjective or an adverb. Next, semantic orientations of these phrases are computed using pointwise mutual information (PMI) method. It is computed by the difference of the mutual information between the feature and the word "excellent" and mutual information between the feature and the word "poor." Finally, overall polarity of the document is recognized by averaging the semantic orientations of all the phrases in the document. If average was positive, document was classified as positive document; otherwise it was classified as negative document.

Turney and Littman [133] introduces the method for inferring the semantic orientation of a word based on the statistical association of a word with the fixed set of positive and negative words. They experimented with two approaches viz. pointwise mutual information (PMI) and latent semantic analysis (LSA). Both PMI an LSA techniques are based on the word co-occurrence with the hypothesis that the semantic orientation of a word tends to correspond to the semantic orientation of the words appearing near to that. On that basis, authors have used PMI method to calculate the semantic orientation of a given word/phrase by comparing its similarity to a positive reference word ("excellent") with its similarity to a negative reference word ("poor"). The semantic orientation of a given word is calculated from the strength of its association with a set of positive words, minus the strength of its association with a set of negative words. Authors have also used LSA technique to calculate the strength of the semantic association between words. LSA is based on the singular value decomposition (SVD); it analyzes the statistical relationships among words in a corpus. LSA technique works as follows to compute the semantic orientation of a word. The first step is to construct a matrix X from the text in which the row vectors represent words and the column vectors represent chunks of text (e.g., sentences, paragraphs, documents). Each cell represents the weight of the word using TF-IDF weighting scheme. The next step is to apply SVD to matrix X, to decompose X into a product of three matrix $U \sum V^T$. U and V are in column orthonormal form. $\sum$ is a diagonal matrix of singular value. X can be approximated by the matrix $U_k \sum_k V_k^T$ by selecting the top k singular values and vectors. The similarity of two words LSA(word1, word2) is measured by the cosine of the angle between their corresponding row vectors of U_k.

Tan et al. [125] proposed a linguistic approach that combines the typed dependencies and subjective phrases to identify the sentence-level sentiment polarity. Their approach considered the intensity of words and domain terms that influence the sentiment polarity in detecting the polarity of the sentence. Tan et al. [126] studied the complex relationships between the words using class sequential rules (CSR) to learn the typed dependency patterns to detect sentiment polarity at the phrase level. Class sequential rules (CSRs) are very similar to association rule mining (ARM). ARM uses the concept of frequent items set; similarly, CSR method considers the subsequence called as sequential patterns and class labels. Authors initially transformed the typed dependency patterns in sequences; and further, class sequential rules are generated based on the subsequences of the dependency patterns found in the sequential rules. Further, with the CSR algorithm, authors extracted unigram and bigram pattern rules based on an experimentally determined minimum support threshold and minimum confidence threshold. Mukras et al. [78] proposed the method for automatic identification of POS-based pattern for extraction of polar phrases. They applied various feature selection methods, namely, information gain (IG), chi-squares (CHI), and document frequency (DF), for identification of important phrase patterns. Further, they applied PMI method to compute the semantic orientation of phrases.

Zhang et al. [147] proposed a method to determine the polarity of the sentence based on word dependency rules and then predicted the polarity of the document by summing up the results from each sentence. Fei et al. [39] constructed phrase patterns with adjectives, adverbs, prepositions, conjunctions, noun, and verbs. Further, they proposed an unsupervised method to determine the semantic orientation of these phrases, and finally overall polarity of the text is determined by summing up the semantic orientation of all the words.

Takamura et al. [121] proposed a method to compute the semantic orientation of words using spin models. They constructed a network of words with the help of gloss definition, thesaurus, and word co-occurrence statistics. They treated each word as an electron, and each electron has a spin, and each spin has two directions that correspond to two values, i.e., up and down. Their method is based on the hypothesis that two neighboring spins tend to have the same direction of spins and also tend to have similar semantic orientation. Takamura et al. [122] proposed latent variable models for predicting the semantic orientation of the phrases. However, their model was limited for the phrases which were seen in the training corpus. Further, Takamura et al. [123] proposed the method for determining the semantic orientation of adjective-noun pair phrases which were able to predict the semantic orientation of unseen phrases.

Sentiment-bearing feature extraction from the unstructured text is important to detect the overall sentiment of the text. Researchers used dependency tree to get the opinionated phrases from the text. Thet et al. [128] generated dependency tree of a sentence and split the sentence into clauses. Further, contextual sentiment score for each clause is determined. Kaji and Kitsuregawa [55] extracted polar sentences from Japanese HTML documents using language structural clues. Next, phrases are extracted from polar sentences using dependency parser. Further, semantic score of each polar phrase is computed using chi-square and PMI method. Qiu et al. [105] proposed a novel semi-supervised, double propagation approach for the opinion lexicon expansion and target extraction problems. They used the relationship between sentiment words and topic or product features to extract new sentiment words. Their extraction rules are based in the dependency rules. They showed that their approach is able to extract a large number of sentiment words.

One of the popular research trends includes the use of the lexical database WordNet [75]. It provides the synsets, i.e., grouping of words into synonym sets and semantic relationship between them such as antonyms, hyponyms, etc. Kamps et al. [56] proposed a method for determining the polarity of a word by its shortest paths to two seed words “good” and “bad” in WordNet. Their proposed idea is similar to the

PMI method of Turney's semantic orientation except that the measure of association is replaced with a measure of semantic distance in WordNet lexicon. Further, Esuli and Sebastiani [37] used WordNet to construct a semantic lexicon having polarity values of words. SentiWordNet provides positive, negative, and objective scores to each gloss, brief description of the synset, in WordNet.

SentiWordNet has been used as a lexicon in many sentiment analysis studies [31, 33, 38]. Ohana and Tierney [85] retrieved semantic orientation scores of adjectives, adverbs, verbs, and nouns from SentiWordNet [37], and further overall polarity of the document is determined by averaging semantic scores of all the words. Fahrni and Klenner [38] proposed an approach to determine target-specific polarity of adjectives. The idea behind their method was that prior polarity of the domain-specific noun is modified by a qualifying adjective. Wikipedia is used in order to detect the automatic target noun, and further a bootstrapping approach is used to determine the target-specific adjective polarity. SentiWordNet is used to determine the prior polarity of an adjective. Hu and Liu [52] used WordNet synonyms and antonyms to determine the semantic orientation of words. Their method bootstraps the polarity value from words with known polarity to the unknown polarity words. They assigned the polarity value to a given word on the basis of the polarity value of its synonyms and antonyms whose polarity value is known in advance. Cambria et al. [24] presented an effective SenticNet API for concept-level sentiment analysis.

ConceptNet is a large semantic network consisting of a large number of commonsense concepts [48, 65]. Commonsense knowledge in ConceptNet is the largest publicly available commonsense knowledge base which can be used to mine various inferences from the text. Recently, researchers have used commonsense knowledge based on ConceptNet for concept-level sentiment analysis [22, 23, 97, 100, 101]. Sureka et al. [119] proposed a method for determining the semantic orientation of subjective words using commonsense knowledge (ConceptNet).

Overall, semantic orientation approaches are on computation of semantic orientation of words based on co-occurrence frequency of the words, and the second is based on the use of sentiment lexicons as WordNet, SentiWordNet, etc. The main drawback of the semantic orientation-based approaches is that opinion words identified may not necessarily contain the accurate polarity. The polarity of the opinion words may vary with the context. In this book, an approach is presented to determine the polarity of words considering the context information. Corpus-based approaches suffer from the necessity of a large corpus to determine the accurate polarity of the words, whereas dictionary-based approaches do not require a large corpus. But, it is challenging to construct a large sentiment lexicon having semantic orientation values for all the domains. In addition, most of the existing sentiment analysis models don't consider the importance of the product feature in determining the overall sentiment of the document. Therefore, a sentiment analysis model is proposed in this book which considers the importance of a feature with respect to the domain with the help of developed ontology.

Table 2.1 presents a summary of some of the important sentiment analysis models.

Table 2.1 Existing sentiment analysis models

Author	Approach/ method/ features	Machine learning algorithm	Dataset	Resources used	Evaluation method	Type	Best accuracy
Pang and Lee [92]	Based on minimum cuts method with subjectivity detection	NB, SVM	Movie reviews	NA	10-fold cross Validation	ML	86.4 %
Pang et al. [91]	Unigrams, bigrams	NB, SVM, MaxEnt	Movie review (700 positive, 700 negative)	NA	3-fold cross validation	ML	82.9
Turney [132]	PMI-IR	NA	Automobile, bank, movie, travel reviews	Search engines	NA	SO	65.8–84
Dave et al. [30]	Unigrams, bigrams, trigrams	Scoring, smoothing, NB, ME, SVM	Product reviews	NA	NA	ML	88.9
Kim and Hovy [59]	Probabilistic based	NA	DUC corpus	NA	10-fold cross validation	ML	81.0
Prabowo and Thelwall [104]	Heuristic rules	Rule, statistical, induction-based classifier, SVM	Movie, product, myspace comments	General inquirer	10-fold cross	ML	87.3
Gamon [42]	Linguistics features	SVM	Customer feedback	NA	10-fold cross validation	ML	77.5
Hiroshi et al. [49]	NLP patterns	Rules	Camera reviews	NA	NA	SO	Recall 43
Mullen and Collier [79]	Hybrid method with Turney and Osgood values	Hybrid SVM	Movie review	WordNet	3-fold cross validation	ML	86
O'keefe and Koprinska [86]	SentiWordNet-based features and feature selection	SVM, NB	Movie review	SentiWordNet	10-fold cross validation	ML	87.15
Tan et al. [126]	Class sequential rules (CSRs)	Heuristic rules	Movie review	Sentiment Lexicon, SentiWordNet	10-fold cross validation	SO	85.87
Ohana and Tierney [85]	Unigrams	Aggregation of polarity values	Movie reviews	SentiWordNet	3-fold cross validation	SO	65.85

(continued)

Table 2.1 (continued)

Author	Approach/ method/ features	Machine learning algorithm	Dataset	Resources used	Evaluation method	Type	Best accuracy
Denecke [31]	Unigrams + polarity scores	Rule-based classifier	MPQA corpus, movie reviews	SentiWordNet	NA	SO	Precision 61 %, recall 68 %
Salvetti et al. [112]	POS-filtered unigrams	NB, Markov model	Movie reviews	WordNet	NA	ML	80.5
Konig and Brill [60]	Pattern-based	Hybrid SVM	Movie reviews, customer feedback data	NA	5-fold cross validation	ML	89.0
Tan and Zhang [124]	Unigram with IG, MI, DF, CHI	Centroid classifier, NB, SVM, KNN	Movie review in Chinese	NA	NA	ML	88.8
Xia et al. [144]	Ensemble features	Ensemble classifier	Movie review, book, DVD, electronics	NA	5-fold cross Validation	ML	88.0, 83.0, 83.8, 86.0
Ng et al. [81]	Various dependency features	SVM	Movie, book review	NA	10-fold cross validation	ML	90.5
Riloff et al. [109]	NLP patterns and Information Gain	SVM	Movie review, MPQA data	NA	3-fold cross validation	ML	82.7
Xia and Zong [143]	Generalized dependency features	NB, SVM	Movie reviews	NA	5-fold cross validation	ML	88.60
Read [108]	Topic, domain, temporal dependency features	SVM, NB	Movie review	NA	10-fold cross validation	ML	81.5
Dang et al. [29]	Selected content free, content-specific and sentiment features	SVM	Book, DVD, electronics	SentiWordNet	10-fold cross validation	ML	78.85, 80.75, 83.75
Ye et al. [145]	Character Ngrams	SVM, NB	Travel domain reviews	NA	3-fold cross validation	ML	86.06
Kennedy and Inkpen [58]	Contextual valence shifters with 34718 features	Term counting and SVM	Movie reviews	General inquirer	10-fold cross validation	ML	86.20
Martineau and Finin [71]	BoW with smoothed delta IDF	SVM	Movie reviews	NA	10-fold cross validation	ML	88.10
Maas et al. [68]	BoW+ additional unlabeled data	LDA, SVM	Movie reviews	NA	10-fold cross validation	SO	88.90
Nguyen et al. [83]	Rating based features+ unigrams+ bigrams+ trigrams	SVM	Movie reviews	NA	10-fold cross validation	ML	91.6

2.3 Conclusions

In this chapter, we presented most of the state-of-art techniques employed for sentiment analysis. Two main approaches have been explored for document-level sentiment analysis, namely, machine learning and semantic orientation approach. Both the techniques have their benefits and drawbacks. Machine learning approaches produce higher accuracy as compared to semantic orientation-based approaches. It is due to the fact that machine learning models depend on the training dataset; thus, it is domain dependent and can learn the patterns more effectively. In semantic orientation approaches, corpus-based techniques require large corpus to decide the semantic orientation of words. These approaches do not perform well because polarity of words changes with domain and context, and there is no such corpus available which can provide accurate polarity of words depending on the domain and context. Dictionary-based approaches are very easy to develop as it does not require any training data to develop learning model or corpus to determine the polarity value. However, dictionary-based approaches require a large polarity lexicon which is critical to develop. The problem with these approaches is the coverage because most of the available polarity dictionaries contain general knowledge which is insufficient to determine the polarity of the documents.

Chapter 3
Machine Learning Approach for Sentiment Analysis

Machine learning algorithms have been widely used for sentiment analysis [66]. The bag-of-words (BoW) representation is commonly used for sentiment analysis [63, 93]. BoW method assumes the independence of words and ignores the importance of semantic and subjective information in the text. All the words in the text are considered equally important. The BoW representation is commonly used for sentiment analysis, resulting into high dimensionality of the feature space. Machine learning algorithms reduce this high-dimensional feature space with the help of feature selection techniques which selects only important features by eliminating the noisy and irrelevant features. Recently, machine learning-based sentiment analysis models are gaining prominence in the field [66].

In this chapter, we explore various feature extraction techniques proposed in the literature and, further, prepare various composite feature sets. We also extract new sentiment-rich bi-tagged features in addition to Turney's pattern [132] that conform to predefined POS patterns. Further, in this chapter, a novel feature extraction technique is also proposed which reduces the feature vector size and also alleviates the data sparseness problem for supervised sentiment analysis using semantic clustering of features. Further, effectiveness of a new feature selection technique, i.e., minimum redundancy maximum relevance (mRMR), is investigated for the sentiment analysis. Finally, support vector machine (SVM) and Boolean multinomial naive Bayes (BMNB) machine learning algorithms are used for classification of documents in positive and negative. Figure 3.1 presents a flow diagram of the proposed machine learning approach for sentiment analysis.

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*, Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_3

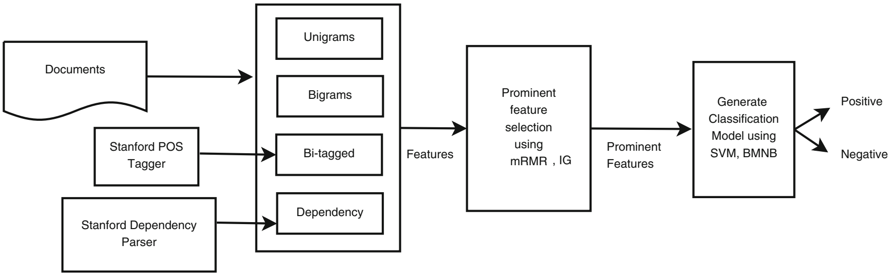

Fig. 3.1 Flow diagram of machine learning approach for sentiment analysis

3.1 Feature Selection Techniques

High dimensionality is a curse for the performance of the machine learning algorithms. There are two types of techniques to reduce the dimensionality of the feature vector length: (1) feature extraction/feature transformation and (2) feature selection [53, 110]. Feature extraction techniques reduce the feature vector length by transformation/projection of all the features in lower-dimensional feature vector. It maps the high-dimensional data on lower-dimensional space. New attributes are obtained by the combination of all the original features, e.g., principal component analysis [53] and singular value decomposition (SVD) [68]. Feature selection techniques select the important features from the high-dimensional feature vector with the help of some goodness of feature, e.g., information gain (IG). It eliminates the irrelevant and noisy features to improve the machine learning algorithms.

Feature selection techniques select the minimum prominent features such that they represent the class attribute in the reduced feature space. Feature selection techniques have two main benefits: firstly, it can significantly improve the classification accuracy, and secondly it provides better insight into prominent class features, resulting in a better understanding of sentiment arguments and features [44]. Reduced feature vector comprising of relevant and prominent features improves the computation speed and increases the accuracy of machine learning methods [1, 50]. Feature selection techniques eliminate noisy and irrelevant features, resulting into more efficient representation of the data in lower feature space. Feature selection methods are basically of two types depending on how they select feature from the feature vector, i.e., filter approach and wrapper approach [53, 110]. In filter approach, all the features are treated independent to each other. Features are ranked according to their importance score, which is calculated by using some function. Filter approach-based methods do not depend on the classifier. Advantages of this approach are that they are computationally simple, fast, and independent to the classifier. Feature selection step is performed once and then reduced feature vector can be used with any classifier. In wrapper approach [53], a search procedure is defined to search the feature subset, and various subsets of features are generated and further evaluated for a specific classifier. In wrapper approach, features are treated dependent to each other, and model interacts with the classifier. As the number of feature subsets grows exponentially with increase in the number of features, hence heuristic search methods are used for selecting feature subsets. The main disadvantage of wrapper approach is that it is computationally expensive. In wrapper approaches, feature subsets depend on the classifier chosen; therefore, it can be different for different classifiers. Filter approach is computationally very efficient as compared to wrapper approach. Wrapper approach becomes complex. It is due to the fact that complexity to create a wrapper for subset selection increases as dimensionality of the feature vector increases.

We also experimented with other feature selection techniques, viz., mutual information and chi-square. The result with these existing feature selection techniques was very low as compared to information gain for sentiment analysis on the same experimental settings. We also experimented with two well-known feature transformation techniques, viz., principal component analysis (PCA) and singular value decomposition (SVD). But the results of these techniques were also very low as compared to information gain technique. This observation has been evident by many researchers in the literature [1, 84, 86, 124, 134]. Since it is observed by many researchers and it is a well-known finding in the NLP community that IG performs best among other feature selection techniques, in this book, only results of IG are used to compare the results of our proposed technique.

3.1.1 Minimum Redundancy Maximum Relevance

The minimum redundancy maximum relevance (mRMR) [94] is a filter-based feature selection technique that is used to select the prominent features of a class. Feature selection is one of the main problems in machine learning; it identifies the subsets of features that are highly correlated and strong enough to identify the class. This is called maximum relevance. These subsets of features generally contain relevant features, but also contain redundant features. mRMR feature selection technique attempts to eliminate these redundant features. This is called minimum redundancy. When two relevant features have redundancy among them, then less important feature can be dropped without compromising the performance of the classifier [94].

mRMR feature selection technique selects the prominent features as follows:

1. Features are selected such that they are highly correlated with the class attribute (maximum relevance).
2. Features are selected such that they are less redundant and still have high correlation with the class attribute (minimum redundancy).

mRMR feature selection technique uses mutual information to measure the correlation among features and class attribute. Mutual information measures the nonlinear correlation between two attributes [69]. Consider feature set $F = \{f_1, f_2, \ldots f_n\}$ containing total n features and class attribute C. The correlation or relevance of a feature f_m with a class attribute C can be expressed in terms of mutual information (joint probability distribution $P(f_m, C)$ and the marginal probability distribution $P(f_m)$, $P(C)$) as shown in Eq. (3.1).

$$A = MI(f_m, C) = \sum_{m,C} P(f_m, C)\log\frac{P(f_m, C)}{P(f_m)P(C)} \tag{3.1}$$

Further, redundant features are eliminated by using the correlation among the features as computed in Eq. (3.2),

$$B = MI(f_i, f_j) = \sum_{i,j} P(f_i, f_j)\log\frac{P(f_i, f_j)}{P(f_i)P(f_j)} \tag{3.2}$$

mRMR feature selection technique selects the features with the help of two schemes as given in Eqs. (3.3) and (3.4), i.e., (i) mutual information difference (MID) and (ii) mutual information quotient (MIQ) scheme. In our experiments, we used MID scheme. Prominent features are selected by maximizing A and minimizing B.

$$\text{MID} = \max(A - B) \tag{3.3}$$

$$\text{MIQ} = \max\left(\frac{A}{B}\right) \tag{3.4}$$

3.1.2 *Information Gain (IG)*

Information gain (IG) is one of the important feature selection techniques for sentiment analysis [1, 69]. IG is used to select important features with respect to class attribute. It is measured by the reduction in the uncertainty in identifying the class attribute when the value of the feature is known. The top-ranked (important) features are selected for reducing the feature vector size and in turn have better classification results. Let class attribute C be total K classes; it can be denoted as $\{C_1, \ldots, C_K\}$. Information gain of a term w can be calculated by using Eq. (3.5).

$$\begin{aligned} \text{IG}(w) = &-\sum_{j=1}^{K} P(C_j)\log(P(C_j)) + P(w)\sum_{j=1}^{K} P(C_j|w)\log(P(C_j|w)) \\ &+ P(\overline{w})\sum_{j=1}^{K} P(C_j|\overline{w})\log(P(C_j|\overline{w})) \end{aligned} \tag{3.5}$$

Here, $P(C_j)$ is the fraction of number of documents that belongs to class C_j out of total documents, and P(w) is the fraction of documents in which term w occurs. $P(C_j|w)$ is computed as the fraction of documents from class C_j that has term w.

3.2 Machine Learning Methods

Machine learning algorithms have been used extensively for sentiment analysis. Performances of various machine learning algorithms have been reported in the literature for sentiment analysis [65, 86, 91, 93]. We also experimented with other popular machine learning algorithms like naive Bayes, MaxEnt, random forest, and neural network for sentiment analysis. We also experimented RBF kernel function (radial basis function) in SVM classifier. But linear SVM classifier significantly outperforms other classifiers for sentiment analysis; this finding resembles with the results of many researchers in the literature [29, 54, 72, 74, 76, 80, 81, 91, 124, 129, 134, 144, 145]. We do not present the results of these experiments in this book because these findings are well known in the NLP community and do not provide any additional information to the reader, and it may also deviate the focus of the book. Since it is observed by many researchers that SVM classifier performs best among other machine learning methods, in this book, we present the results of BMNB classifier, whose performance has not been examined for sentiment analysis [67, 86, 91]. Further, we compare its results with well-known SVM classifier.

3.2.1 Multinomial Naive Bayes

Naive Bayes is a machine learning algorithm, which is frequently used for text classification [69]. It is computationally very efficient and easy to use. The naive Bayes classifier treats the features conditionally independent of each other, given the class. Multinomial naive Bayes with term frequency is a variant of NB classifier, denoted as TMNB. It is a probability-based learning method, which constructs a model by using the term frequency of a feature/word/term to compute the probability [69]. In naive Bayes, class probability for a given document is computed, and document is assigned in the class which has highest probability. The Boolean multinomial naive Bayes (BMNB) is the same TMNB, except that the term frequency of a word in a document is counted as 1 if a term is present else it is counted as 0. Training and testing algorithms for BMNB classifier are given in Algorithms 3.1, 3.2 [69]. In text classification, the goal is to find the best class for the given document $d = \{t_1, t_2, \ldots, t_y\}$. The best class in NB classification is the most likely or maximum a posteriori (MAP) class c_{map} as shown in Eq. (3.6).

$$C_{\mathrm{map}} = \arg\max_{c \epsilon C} P(c) \prod_{i \epsilon y} P(t_i|c) \tag{3.6}$$

Here, $P(t_i|c)$ is the conditional probability of term t_i occurring in a document of class C; it can be computed as shown in Eq. (3.8). We interpret $P(t_i|c)$ as a measure of how much evidence t_i contributes that C is the correct class. $P(c)$ is the prior probability of a document occurring in class C; it can be computed as shown in Eq. (3.7) [69].

Algorithm 3.1 Training: Boolean Multinomial Naive Bayes

INPUT labeled document corpus(D): *Positive_documents* (PD) = $\{d_1, d_2, d_3, \ldots d_p\}$, *Negative_documents* (ND) = $\{d_1, d_2, d_3, \ldots d_n\}$, and Class Attribute (C) = {positive, negative}

V = ExtractUniqueTerms(D) // V is vector of all the Bag-of-Words in the corpus
N = CountTotalDocuments(D) // total number of documents in the corpus
// c is the member of class C.
for each $c \in C$ **do**
  $N_c = CountTotalDocumentsInClass(D, c)$
  $Prob[c] = \frac{N_c}{N}$
end for
$n_{t,c} = CountDocumentsInClass(t, D)$ //count total documents in class c where term t appears
for each term $t \in V$ **do**
  $P(t|c) = \frac{n_{t,c}+1}{N_c+|V|}$
end for
return $V, Prob[c], P(t|c)$

$$P(c) = \frac{N_c}{N} \tag{3.7}$$

where N is the total number of documents in the corpus and (N_c) is the number of documents that belongs to class c.

$$P(t|c) = \frac{n_{t,c} + 1}{N_c + |V|} \tag{3.8}$$

where $(n_{t,c})$ is total documents in class c, where the term t appears.

Algorithm 3.2 Testing: Boolean Multinomial Naive Bayes

INPUT Classes (C),given document (d), Document corpus (D), Vocabulary (V), Probability of class $Prob[c]$, Conditional probability $P(t|c)$

W = ExtractTermsFromDoc(d) // all the unique bag-of-words in the corpus
T = CountTotalDocuments(D) // total number of documents in the corpus
for each $c \in C$ **do**
  $score[c] = logProb[c]$
  for each $t \in W$ **do**
    $score[c] += logP(t|c)$
  end for
end for
$decision = \arg\max_{c \in C} score[c]$
return $decision$

3.2.2 Support Vector Machine

Support vector machine (SVM) is a supervised learning method [69]. SVM transforms the original training data into a higher-dimensional space so that data point can be linearly separated. Further, within this new dimension, it searches for the linear optimal separating hyperplane that is used for identifying the class of a new incoming testing sample. The main objective of SVM model is to find an optimal hyperplane that separates the input training data in such a way that one category of data variables is on one side of the hyperplane and the other category of the data variables is on the other side of the hyperplane. SVM finds a hyperplane that divides the training documents in such a way that both the class data points are maximum separable. SVM has shown to be superior in comparison to other machine learning algorithms, in case of limited but sufficient training samples. SVM has been widely used for text classification and sentiment analysis [72, 91].

3.3 Feature Extraction Methods

Various feature sets are extracted for sentiment analysis to investigate the contribution of each type of feature for sentiment analysis. Initially, four types of basic features are extracted, namely, unigrams (F1 feature set), bigrams (F2 feature set), bi-tagged (F3 feature set), and dependency parsing tree-based features (F4 feature set). Next, prominent feature sets are constructed by eliminating noisy and irrelevant features from the basic feature sets using feature selection techniques. Information gain (IG) and mRMR feature selection methods are used to select prominent features. Further, various composite feature sets are produced using basic and prominent feature sets. The process of formation of all these feature sets is described in subsequent subsections in detail and also presented in Fig. 3.2.

3.3.1 Basic Features

3.3.1.1 Unigrams and Bigrams

Unigram features are simply bag-of-words (BoW) features extracted by eliminating extra spaces and noisy characters between two words. An example is the sentence “this is an awesome movie.” Here, words “this,” “is,” “an,” “awesome,” and “movie” are all unigram features. Bigrams are the features, consisting of every two consecutive words in the text. For the above example, “this is,” “is an,” “an awesome,” and “awesome movie” are the bigram features. These features are capable of including some contextual information.

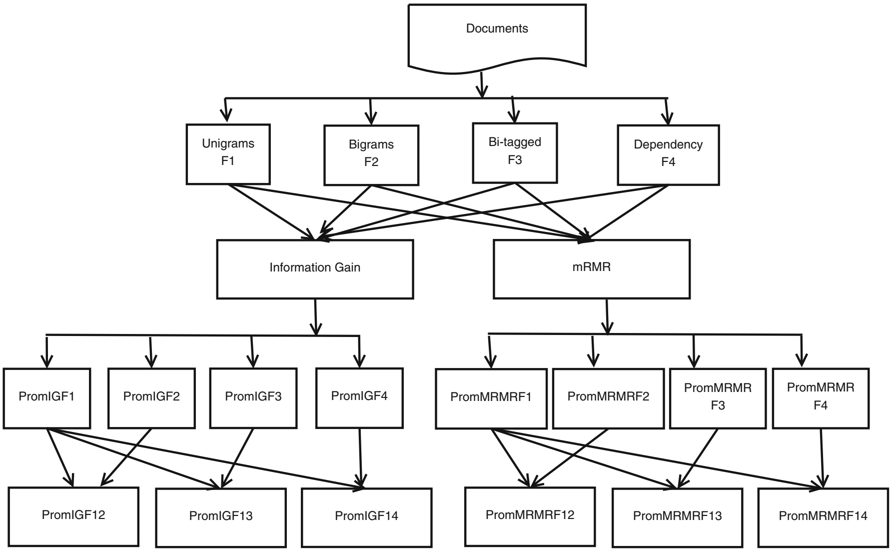

Fig. 3.2 Process of formation of various feature sets

3.3.1.2 Bi-tagged

Bi-tagged features are selectively extracted using part-of-speech (POS)-based fixed patterns [8]. Bigrams containing mostly adjectives and adverbs are considered more sentiment bearing. POS-based information is used to extract the sentiment-rich features, as it has been investigated in literature that adjective and adverbs are subjective in nature [47, 132]. Turney [132] proposed a method to extract two-word sentiment-rich features such that its one member is either adjective or adverb (i.e., adjective-noun, adverb-adjective, adjective-adjective, noun-adjective, adverb-verb). However, it is observed that verbs can also contain sentiment information that is useful for sentiment analysis (i.e., verb-noun, verb-adjective, adjective-verb, adverb-adverb). Thus, we extend the rule set to extract more sentiment-bearing features. For example, "*like_VB movie_NN*," "*feel_VBP good_JJ*," etc., these sentiment-rich features were not extracted with Turney's rule set. The JJ/JJR/JJS tags indicate adjectives, the NN/NNS tags are nouns, the RB/RBR/RBS tags are adverbs, and the VB/VBG/VBP tags are verbs. Stanford POS tagger is used for tagging words in the sentence.

3.3.1.3 Dependency Features

An in-depth linguistic analysis in the form of syntactic relations among the words in a sentence can be useful for sentiment analysis model. It is investigated in the literature that syntactic patterns are very effective for subjective detection [72, 80, 89]. Dependency parsing tree is used to capture long-distance information in the text. The Stanford Parser is used to construct dependency parsing tree [70].

For example, if the sentence is "the movie sounds boring," POS-tagged sentence is as follows: "the_DT movie_NN sounds_VBZ boring_JJ," and dependency relations are as follows: det (movie-2, the-1), nsubj(sounds-3, movie-2), root(ROOT-0, sounds-3), and xcomp(sounds-3, boring-4). Here, dependency features are movie_the, sounds_movie, sounds_boring.

3.3.2 Prominent Features

Feature selection techniques are important for improving the performance of the machine learning method in terms of accuracy as well as execution time. It can select prominent features and eliminate the noisy and irrelevant features that deteriorate the efficiency of the classifiers. Information gain (IG) and mRMR feature selection methods are used in our experiments independently.

Initially, information gain (IG) feature selection technique is used to extract prominent features. Since IG is a feature selection technique based on threshold value, in our experiments, all the features are considered prominent feature whose IG value is greater than 0, and all the features are dropped whose IG value is equal to 0. Prominent features extracted using IG as feature selection method from unigrams, bigrams, bi-tagged phrases, and dependency features are named as PromIGF1 (prominent unigrams), PromIGF2 (prominent bigrams), PromIGF3 (prominent bi-tagged), and PromIGF4 (prominent dependency) features, respectively. Also, mRMR feature selection technique is used to extract important features from the feature vector. mRMR feature selection method ranks the features according to their importance for classification. Top-ranked features are selected for sentiment analysis. Prominent features extracted using mRMR as feature selection technique from unigrams, bigrams, bi-tagged phrases, and dependency features are named as PromMRMRF1 (prominent unigrams), PromMRMRF2 (prominent bigrams), PromMRMRF3 (prominent bi-tagged), and PromMRMRF4 (prominent dependency) features, respectively. Various prominent feature vectors are presented in Table 3.1.

3.3.3 Composite Features

Performance of unigrams increases when combined with bigrams but at a high computational cost due to increased feature vector length [81, 91]. We also investigate the performance of various composite features. Composite features are

Table 3.1 Various prominent features sets

Features	Feature construction method
PromIGF1	Prominent unigrams features using IG as feature selection
PromIGF2	Prominent bigrams features using IG as feature selection
PromIGF3	Prominent bi-tagged features using IG as feature selection
PromIGF4	Prominent dependency features using IG as feature selection
PromMRMRF1	Prominent unigrams features using mRMR as feature selection
PromMRMRF2	Prominent bigrams features using mRMR as feature selection
PromMRMRF3	Prominent bi-tagged features using mRMR as feature selection
PromMRMRF4	Prominent dependency features using mRMR as feature selection

Table 3.2 Various composite feature sets

Feature set	Feature set construction method
ComF12	Composite feature set with unigrams and bigrams
ComF13	Composite feature set with unigrams and bi-tagged features
ComF14	Composite feature set with unigrams and dependency features
PromIGF12	Composite feature set with prominent IG unigrams and bigrams
PromIGF13	Composite feature set with prominent IG unigrams and bi-tagged features
PromIGF14	Composite feature set with prominent IG unigrams and dependency features
PromMRMRF12	Composite feature set with prominent mRMR unigrams and bigrams
PromMRMRF13	Composite feature set with prominent mRMR unigrams and bi-tagged features
PromMRMRF14	Composite feature set with prominent mRMR unigrams and dependency features

created using unigram with bigrams named as ComF12, unigrams with bi-tagged named as ComF13, and unigrams with dependency features named as ComF14 features. Various composite feature vectors are presented in Table 3.2.

By IG and mRMR feature selection techniques, prominent feature sets are constructed and further combined to get the composite feature sets. PromIGF12 feature set is generated by combining the prominent IG unigrams (PromIGF1) and prominent IG bigrams (PromIGF2). PromIGF13 feature set is created by combining prominent IG unigrams (PromIGF1) and prominent IG bi-tagged features (PromIGF3). Similarly, PromIGF14 feature set is created with prominent IG unigrams (PromIGF1) and prominent IG dependency features (PromIGF4).

Similarly, various composite feature sets are constructed by combining prominent mRMR features. PromMRMRF12 feature set is created by combining prominent mRMR unigrams (PromMRMRF1) and prominent mRMR bigrams (PromMRMRF2). PromMRMRF13 feature set is constructed by combining prominent mRMR unigrams (PromMRMRF1) and prominent mRMR bi-tagged features (PromMRMRF3), and further, PromMRMRF14 feature set is generated by combining prominent mRMR unigrams (PromMRMRF1) and prominent mRMR dependency features (PromMRMRF4).

3.3.4 Clustering Features

Performance of machine learning models relies on the quality of the features passed to it and the feature vector size. Feature vector size is the number of features in a feature vector. The quality of a feature can be measured with its discriminating ability for the classes [44]. The feature selection techniques have been reported in the literature to select only important features on the basis of discriminating ability of the feature, i.e., information gain (IG), mutual information (MI), etc. [3, 93, 124]. Feature selection techniques select important features by dropping less important features. The problem with existing feature selection techniques is that the less important features are dropped. But dropping a feature means losing some information, resulting into unseen words in the training data. Words/terms appeared in the test data but not in the training data, degrading the performance of the machine learning algorithm. Hence, if feature selection technique is used, some information may be lost and classifier may suffer from unseen data. In contrast, if feature selection technique is not used, classifier may suffer from the sparseness of data and high-dimensional feature vector problem.

In the proposed method [9], we do not drop any feature from the training data. We create clusters of features having similar semantic orientation. The advantage of the proposed method is that classifier is not dependent much on the words/terms chosen by the authors to express their opinion, as authors may choose different words for expressing the same opinion. For example, authors can use "awesome," "good," and "nice" for expressing similar positive sentiment. Therefore, we cluster the words having similar sentiment into one cluster, and that is considered a single feature for classification. In contrast, a simple feature selection technique may drop some features, those that can result into unseen features, which may degrade the performance of the classifier.

The proposed method to construct feature vector with semantic clustering is presented in Algorithm 3.3. Semantic clustering feature set is constructed as follows: firstly, unique words/features are extracted from the review documents. Then, all the words having similar semantic orientation are grouped together to form a cluster. Semantic orientation of the features is computed using pointwise mutual information (PMI). Finally, binary weighting scheme is used to assign weights to the features, i.e., weight is assigned 1 (one) if a word is present in the cluster else weight is assigned 0 (zero).

The range of semantic orientation values is from -1 to 1. Clusters are named as $\{cluster_{(-100)}, cluster_{(-99)}, \ldots . cluster_{(-1)}, cluster_0, cluster_1, cluster_2 \ldots cluster_{(100)}\}$ to generate a feature vector with 201 clusters. Similarly, if total number of clusters to be created are 401, so clusters are named as $\{cluster_{(-200)}, cluster_{(-199)}, \ldots . cluster_{(-1)}, cluster_0, cluster_1, cluster_2 \ldots . cluster_{(200)}\}$. Each cluster represents the degree of sentiment being expressed, for example, $cluster_{55}$ is a collection of words whose semantic orientation is 0.55.

Consider an example to understand the construction of feature vector of size 201; if a document has 5 words, i.e., word1, word2, word3, word4, and word5, and their semantic orientation values are 0.23, 0.43, 0.51, 0.53, and 0.77, respectively, then feature vector for this document would have value 1 for $cluster_{23}$, $cluster_{43}$, $cluster_{51}$, $cluster_{53}$, and $cluster_{77}$, and all other clusters would have value 0.

Algorithm 3.3 Feature vector generation with semantic clustering

INPUT labeled document corpus D = $\{d_1, d_2, d_3, \ldots d_m\}$, number of clusters = n
OUTPUT Feature vector of the size of clusters

T = ExtractUniqueTerms(D) // all the unique bag-of-words are extracted
F = CountTotalUniqueTerms(T) // count the total number of unique terms in the corpus
P = TotalPositiveDocuments(D) // count the total number of positive documents
N = TotalNegativeDocuments(D) // count the total number of negative documents
f_p = frequency of a term in positive documents (f_p = 0) // initialize the variable f_p to 0.
f_n = frequency of a term in negative documents (f_n = 0) // initialize the variable f_n to 0.
Threshold (th) = ($2/n$)
$Feature\_vector = \{cluster_{(-n/2)}, cluster_{(-n/2-1)}, \ldots\ldots.. cluster_{-1}, cluster_0, cluster_1, cluster_2, \ldots\ldots cluster_{(n/2)}\}$
// count the frequency of a term in positive and negative documents
for Each $t_i \in T$ **do**
  for Each $d_i \in D$ **do**
    if $t_i \in d_i$ && label of d_i is positive **then**
      $f_p = f_p + 1$
    else
      $f_n = f_n + 1$
    end if
  end for
end for
//count the semantic orientation of each term.
for Each $t_i \in T$ **do**
  $SO(t_i) = \log \frac{(fp*N)}{(P*fn)}$
end for
//round off the semantic orientation of each term and assign in the corresponding cluster.
for Each term t_i **do**
  P_i = floor(SO(t_i)*th)
  assign term ti in Cluster(P_i)
end for
//Construct the feature vector for each document.
for each document D_i **do**
  for each t_i in D_i **do**
    if t_j is present in $Cluster_j$ **then**
      $Weight_j = 1$
    else
      $Weight_j = 0$
    end if
  end for
end for

Computation of semantic orientation of the feature is based on the assumption that if a feature is occurring frequently and predominantly in one class (positive or negative), then that feature would have high polarity. If a feature has high positive polarity value, it indicates that the feature has occurred mostly in positive documents. Pointwise mutual information (PMI) is used to calculate the strength of association between a feature and positive or negative documents in sentiment analysis [132]. Equation 3.9 shows the strength of association between a feature c with positive documents, and Eq. 3.10 shows the strength of association between a feature c and negative documents [55].

$$\mathrm{PMI}(c, \mathrm{pos}) = \log_2 \frac{P(c, \mathrm{pos})}{P(c)P(\mathrm{pos})} \tag{3.9}$$

$$\mathrm{PMI}(c, \mathrm{neg}) = \log_2 \frac{P(c, \mathrm{neg})}{P(c)P(\mathrm{neg})} \tag{3.10}$$

Here, $P(c, \mathrm{pos})$ is a probability of a feature c that occurs in positive documents, i.e., frequency of the positive documents in which feature c occurs divided by a total number of positive documents. $P(c, \mathrm{neg})$ is the probability that a feature c occurs in negative documents, i.e., frequency of negative documents in which feature c occurs divided by a total number of negative documents. Polarity value of the feature is determined by their PMI value difference [132]. Semantic orientation (SO) of a feature c is computed using Eq. (3.11).

$$\mathrm{SO}(c) = \mathrm{PMI}(c, \mathrm{pos}) - \mathrm{PMI}(c, \mathrm{neg}) \tag{3.11}$$

3.4 Dataset, Experimental Setup, and Results

3.4.1 Dataset Used

To evaluate the performance of the proposed methods, one of the most popular publicly available movie review datasets is used [92]. This standard dataset, known as Cornell Movie Review Dataset, consists of 2000 reviews containing 1000 positive- and 1000 negative-labeled reviews collected from Internet Movie Database (IMDb). To make the experiments scientifically more stable, product review datasets are also used that consist of Amazon product reviews. This benchmark dataset [18] consists of various domain reviews. We use product reviews of books, DVD, and electronics domains to evaluate the performance of all the proposed methods. Each dataset has 1000 positive- and 1000 negative-labeled reviews. More details about the datasets, viz., minimum, maximum, and average word length in a document and a total number of words in the document corpus, are mentioned in Table 3.3. Dataset statistics presented in Table 3.3 shows that movie review dataset contains long reviews, whereas electronics dataset contains short reviews.

Table 3.3 Dataset statistics

	Movie			Book			DVD			Electronics		
	Neg	Pos	Total	Neg	Pos	Total	Neg	Pos	Total	Neg	Pos	Total
Total documents	1000	1000	2000	1000	1000	2000	1000	1000	2000	1000	1000	2000
Min words in a document	16	134	16	5	4	4	11	9	9	9	8	8
Max words in a document	2253	2755	2755	6142	3339	6142	1369	4510	4510	1690	819	1690
Average words in a document	721	803	762	193	168	181	169	170	169	104	99	101
Total words in all documents	721,257	803,117	1,524,374	193,580	168,508	362,088	169,117	170,422	339,539	104,938	98,706	203,644

3.4.2 Evaluation Metrics

During classification process of test samples, certain evaluation parameters are used. A few of these basic parameters, *True Positives*, *True Negatives*, *False Positives*, and *False Negatives*, are described below:

- *True Positives* (*TP*): This is the number of samples belonging to a particular class and correctly identified as such.
- *True Negatives* (*TN*): This is the number of samples not belonging to a particular class and correctly classified as such.
- *False Positives* (*FP*): This is defined as the number of samples belonging to another class but incorrectly classified as belonging to the class being tested.
- *False Negatives* (*FN*): This is the number of samples belonging to the class being tested but incorrectly classified as belonging to another class.

Precision, recall, and F-measure are used for evaluating performance of sentiment analysis [69]. Precision for a class C is the fraction of a total number of documents that are correctly classified and a total number of documents that are classified to the class C (sum of True Positives (TP) and False Positives (FP)) as given in Eq. (3.12). Recall is the fraction of a total number of correctly classified documents to the total number of documents that belongs to class C (sum of True Positives (TP) and False Negative (FN)) as given in Eq. (3.13).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.12}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.13}$$

F-measure is the combination of both precision and recall given by Eq. (3.14).

$$F - \text{measure} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{3.14}$$

F-measure is used to report the performance of classifiers for the sentiment analysis.

3.4.3 Results and Discussions

Support vector machine (SVM) algorithm has been used extensively and reported the best classifier in literature for sentiment analysis [86, 91, 124]. Therefore, we use SVM classifier for classification of review documents in positive or negative class. In addition, we use naive Bayes and Boolean multinomial naive Bayes (BMNB) classifiers. Tenfold cross-validation technique is used to evaluate the performance of the proposed methods; we randomly divide the dataset into 90 % training and

10 % testing documents, such that both the sets are disjoint. We repeat all the experiments 10 times with randomly selected training and testing sets, and final performance is reported by average of the results. Linear SVM, naive Bayes, and naive Bayes multinomial are used for all the experiments with default setting in WEKA [45].

Negation word (no, not, never, didn't, don't, can't) reverses the polarity of the sentence; that is why it is important to handle the negation for sentiment analysis. In our experiments, we adopt a simple approach commonly used by various researchers [91]. It is done by adding *NOT_* to every word between negation word and the first punctuation mark after the negation word. For example, "this is not a good movie," polarity of word "good" is reversed by "not." After negation handling, this sentence becomes "this is *NOT_a NOT_good NOT_movie*." Binary weighting scheme is used to assign the weights to the features as it has been reported better scheme than term frequency-based weighting schemes as [91].

3.4.3.1 Determination of Prominent Features

It is observed from the experiments that among the basic features, unigram features (F1) perform better than other basic features for all the datasets as shown in Table 3.7. For example, unigram feature set produces the F-measure 82.7 % as compared to 79.2 %, 76.4 %, and 78.1 % for bigrams, bi-tagged, and dependency features, respectively, with BMNB classifier on movie review dataset as shown in Table 3.7. The possible reasons for this observation are described as follows. Firstly, bigram feature set comprises of lots of noisy features, which deteriorates the performance of classification. In addition to this, bigram feature set is sparser than unigrams in nature that degrades the performance. Next, bi-tagged features are polar in nature and important for sentiment analysis, but some important features are left. Bi-tagged feature set extracts important but less information which is not sufficient for sentiment analysis and hence performs worst if used independently. Dependency features face the problem of overgeneralization that means features extracted from dependency rules are present in both the classes, resulting into less discriminating features. That is why these features do not contribute much for classification. For example, "this is not a very good movie" is a negative review and "this is a good movie" is a positive review. In this example, both the reviews contain many common features like "movie_good," "movie_is," and "movie_this." These common features are less discriminating features for classification. Lastly, two or more word features face the problem of sparsity unlike in unigrams.

Tables 3.7 and 3.8 present the results with respect to F-measure for all the basic features and prominent features, whereas Table 3.9 presents results with respect to F-measure for various composite features. Tables 3.4 and 3.5 present the feature vector length of all the basic and prominent feature sets used by machine learning methods for classification. Performance of unigram features is considered as baseline in all the experiments [58, 72, 82, 83, 92, 131, 139, 144].

Table 3.4 Feature vector size for basic features with various datasets

S.No	Features	Movie review	Book	DVD	Electronics
1	$F1$	9045	5391	5955	4270
2	$F2$	6050	6484	8888	5513
3	$F3$	4841	3013	4510	2560
4	$F4$	7010	6110	7500	6010

Table 3.5 Feature vector size for prominent features with various datasets

S.No	Features	Movie review	Book	DVD	Electronics
1	$PromIGF1$, $PromMRMRF1$	1130	1152	946	612
2	$PromIGF2$, $PromMRMRF2$	1420	547	995	480
3	$PromIGF3$, $PromMRMRF3$	1114	302	425	295
4	$PromIGF4$, $PromMRMRF4$	1050	430	485	315

Next, prominent feature sets are constructed from the basic features with the help of IG and mRMR feature selection methods. For all the prominent features, performance of all the basic features increased significantly for all the datasets. For example, PromIGF1 features increased the F-measure from 82.7 % to 87.3 % (+5.5 %) with BMNB classifier for movie review dataset. It is due to the fact that feature selection methods improve the performance by eliminating noisy and irrelevant features. Between both the feature selection methods (i.e., IG and mRMR), mRMR methods performed better than IG for most of the feature sets with both the classifiers (i.e., SVM, BMNB), results as shown in Table 3.8. For example, prominent unigrams with mRMR (PromMRMRF1) produced F-measure of 88.1 % as compared to its equivalent feature set (PromIGF1), i.e., 87.3 %. It is due to the fact that mRMR feature selection method can eliminate the noisy, irrelevant, and redundant features and selects only relevant features unlike IG which only selects relevant features and doesn't eliminate redundant features. Therefore, mRMR feature selection method selects more relevant and discriminating features by reducing redundancy.

It is clear from the experiments that unigram features performed better than other features independently. Therefore, composite features are constructed by including extra information with unigrams. Contextual, part-of-speech (POS), and syntactic informations are included with unigrams with the help of bigrams, bi-tagged, and dependency features. Experimental results showed that performances of all the composite features are increased from the baseline unigram features for all the four datasets (results given in Table 3.9). However, execution overhead is also increased due to increased feature vector length. The main reason for this observation is that features add more information which include more language-related information unlike unigrams which is just a bag-of-words approach. However, due to increased feature vector length, computational cost has also increased. Among composite features (i.e., ComF12, ComF13, and ComF14), ComF12 performed better as compared to other composite features with both the classifiers on all the four

Table 3.6 F-measure (in %) for varying cluster size with BMNB and SVM classifiers for all the four datasets (Partly results were initially published in Agarwal B., Mittal N., "Semantic Feature Clustering for Sentiment Analysis of English Reviews", In IETE Journal of Research, Taylor Francis, Vol: 60 (6), 14 Oct, 2014, pp: 414–422, reprinted by permission of the publisher (Taylor & Francis Ltd, http://www.tandfonline.com))

	Movie		Book		DVD		Electronics	
Clusters	BMNB	SVM	BMNB	SVM	BMNB	SVM	BMNB	SVM
101	76.2 (−7.8 %)	74.5 (−11.5 %)	73.7 (−5.9 %)	71.2 (−6.5 %)	74.9 (−5.0 %)	73.2 (−5.3 %)	72.7 (−10.0 %)	75.9 (−0.7)
201	85.4 (+3.2 %)	80.5 (−4.3 %)	84 (+7.1 %)	82.5 (+8.2 %)	83.2 (+5.4 %)	80.2 (+3.7 %)	82.4 (+1.9 %)	78.5 (+2.6 %)
301	88.4 (+6.8 %)	85.6 (+1.6 %)	**87.9 (+12.1 %)**	**88.2 (+15.7 %)**	86.1 (+9.1 %)	85.7 (+10.8 %)	**88.4 (+9.4 %)**	84.3 (+10.1 %)
401	89.1 (+7.7 %)	**88.5 (+5.1 %)**	87.5 (+11.6 %)	87.1 (+14.3 %)	**87.9 (+11.4 %)**	**86.8 (+12.2 %)**	87.9 (+8.7 %)	**87.8 (+14.7 %)**
801	**89.2 (+7.8 %)**	88.4 (+4.9 %)	86.4 (+10.2 %)	86.2 (+13.1 %)	86.2 (+9.2 %)	83.5 (+8.0 %)	84.1 (+4.0 %)	86.4 (+12.9 %)
2001	87.6 (+5.9 %)	85.1 (+1.0 %)	80.4 (+2.5 %)	79.4 (+4.1 %)	79.3 (+0.5 %)	76.2 (−1.4 %)	78.8 (−2.4 %)	80.2 (+4.8 %)
4001	81.9 (−0.9 %)	82.1 (−2.4 %)	78 (−0.5 %)	75.9 (−0.3 %)	78.5 (−0.5 %)	77 (−0.3 %)	80.7 (−0.1 %)	76.4 (−0.1 %)
8001	82.7 (+0.0)	84.2 (+0.0)	78.4 (+0.0)	76.2 (+0.0)	78.9 (+0.0)	77.3 (+0.0)	80.8 (+0.0)	76.5 (+0.0)

Table 3.7 F-measure (in %) for basic features sets (Agarwal B., Mittal N., "Optimal Feature Selection for Sentiment Analysis", In Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2013), Vol: 7817, No: 1, March 24, 2013, pp: 13–24, reprinted with the permission of the publisher (Springer, http://link.springer.com/) and Agarwal B., Mittal N., "Semantic Feature Clustering for Sentiment Analysis of English Reviews", In IETE Journal of Research, Taylor Francis, Vol: 60 (6), 14 Oct, 2014, pp: 414–422, reprinted by permission of the publisher (Taylor & Francis Ltd, http://www.tandfonline.com))

	Movie			Book			DVD			Electronics		
	BMNB	SVM	NB	BMNB	SVM	NB	BMNB	SVM	NB	BMNB	SVM	NB
Unigrams (F1)	**82.7**	**84.2**	**79.4**	**78.4**	**76.2**	**74.5**	**78.9**	**77.3**	**75.4**	**80.8**	**76.5**	**74.9**
Bigrams (F2)	79.2 (−4.2 %)	78.8 (−6.4 %)	76.2 (−4.0 %)	66.8 (−14.7 %)	69.5 (−8.7 %)	67.3 (−9.6 %)	67.1 (−14.9 %)	68.0 (−12.0 %)	67.2 (−10.8 %)	72.6 (−10.1 %)	70.1 (−8.3 %)	71.0 (−5.2 %)
Bi-tagged (F3)	76.4 (−7.6 %)	75.3 (−10.5 %)	74.7 (−5.9 %)	69.9 (−10.8 %)	65.5 (−14.7 %)	68.1 (−8.5 %)	68.9 (−12.0 %)	69.0 (−10.7 %)	66.4 (−11.9 %)	73.2 (−9.4 %)	69.9 (−8.6 %)	69.8 (−6.8 %)
Dependency features (F4)	78.1 (−5.5 %)	77.4 (−8.0 %)	75.4 (−5.0 %)	70.9 (−9.5 %)	70.3 (−8.1 %)	67.5 (−9.3 %)	70.9 (−10.1 %)	69.2 (−10.4 %)	68.7 (−8.8 %)	70.7 (−12.5 %)	69.2 (−9.5 %)	70.3 (−6.1 %)

Table 3.8 F-measure (in %) for various prominent features sets (Agarwal B., Mittal N., "Optimal Feature Selection for Sentiment Analysis", In Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2013), Vol: 7817, No: 1, March 24, 2013, pp: 13–24, reprinted with the permission of the publisher (Springer, http://link.springer.com/) and Agarwal B., Mittal N., "Semantic Feature Clustering for Sentiment Analysis of English Reviews", In IETE Journal of Research, Taylor Francis, Vol: 60 (6), 14 Oct, 2014, pp: 414–422, reprinted by permission of the publisher (Taylor & Francis Ltd, http://www.tandfonline.com))

	Movie			Book			DVD			Electronics		
	BMNB	SVM	NB	BMNB	SVM	NB	BMNB	SVM	NB	BMNB	SVM	NB
PromIGF1	87.3 (+5.5 %)	85.8 (+1.9 %)	85.1 (+7.1 %)	**83.3 (+6.2 %)**	**82.2 (+7.8 %)**	76.3 (+2.4 %)	83.5 (+5.8 %)	81.5 (+5.4 %)	79.2 (+5.0 %)	82.6 (+2.2 %)	81.1 (+6.0 %)	78.3 (+4.5 %)
PromIGF2	81.1 (−1.9 %)	80.4 (−1.4 %)	78.9 (−0.6 %)	71.5 (−8.8 %)	70.5 (−7.4 %)	68.2 (−8.4 %)	75.8 (−3.9 %)	76.1 (−1.5 %)	71.3 (−5.4 %)	79.2 (−1.9 %)	74.9 (−2.0 %)	74.2 (−0.9 %)
PromIGF3	84.3 (+1.9 %)	84.5 (+0.2 %)	80.3 (+1.1 %)	73.1 (−6.7 %)	70.5 (−7.4 %)	70.1 (−5.9 %)	73.8 (−6.4 %)	75.1 (−2.8 %)	74.7 (−0.9 %)	79.1 (−2.1 %)	75.3 (−1.5)	75.8 (+1.2 %)
PromIGF4	81.4 (−1.5 %)	81.1 (−3.6 %)	79.8 (+0.5 %)	75.9 (−3.1 %)	74.7 (−1.9 %)	69.8 (−6.3 %)	74.4 (−5.7 %)	75.8 (−1.9 %)	73.4 (−2.6 %)	77.5 (−4.0 %)	75.1 (−1.8)	74.4 (−0.6 %)
PromMRMRF1	**88.1 (+6.5 %)**	**87.1 (+3.4 %)**	85.4 (+7.5 %)	83.0 (+5.8 %)	82.1 (+7.7 %)	78.9 (+5.9 %)	**84.0 (+6.4 %)**	**82.1 (+6.2 %)**	80.4 (+6.6 %)	**84.2 (+4.2 %)**	**82.8** (+8.2)	78.6 (+4.9 %)
PromMRMRF2	80.1 (−3.1 %)	81.4 (−3.3 %)	79.4 (+0.0 %)	77.2 (−1.5 %)	76.0 (−0.3 %)	70.3 (−5.6 %)	77.8 (−1.3 %)	75.5 (−2.3 %)	72.1 (−4.3 %)	80.2 (−0.7 %)	76.0 (−0.6 %)	73.8 (−1.4 %)
PromMRMRF3	85.1 (+2.9 %)	84.9 (+0.8 %)	81.2 (+2.2 %)	73.7 (−5.9 %)	75.1 (−1.4 %)	71.9 (−3.4 %)	76.2 (−3.4 %)	75.2 (−2.7 %)	75.2 (−0.2 %)	81.3 (+0.6 %)	77.5 (+1.3)	74.2 (−0.9 %)
PromMRMRF4	82.1 (−0.7 %)	81.9 (−2.7 %)	80.9 (+1.8 %)	78.9 (+0.6 %)	76.2 (0.0 %)	70.8 (−4.9 %)	75.2 (−4.6 %)	76.9 (−0.5 %)	73.1 (−3.0 %)	80.1 (−0.8 %)	75.6 (−1.1)	73.9 (−1.3 %)

Table 3.9 F-measure (in %) for various composite feature sets (Agarwal B., Mittal N., "Optimal Feature Selection for Sentiment Analysis", In Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2013), Vol: 7817, No: 1, March 24, 2013, pp: 13–24, reprinted with the permission of the publisher (Springer, http://link.springer.com/) and Agarwal B., Mittal N., "Semantic Feature Clustering for Sentiment Analysis of English Reviews", In IETE Journal of Research, Taylor Francis, Vol: 60 (6), 14 Oct, 2014, pp: 414–422, reprinted by permission of the publisher (Taylor & Francis Ltd, http://www.tandfonline.com))

	Movie			Book			DVD			Electronics		
	BMNB	SVM	NB	BMNB	SVM	NB	BMNB	SVM	NB	BMNB	SVM	NB
ComF12	87.0 (+5.1 %)	86.7 (+2.9 %)	85.4 (+7.5 %)	82.6 (+5.3 %)	79.5 (+4.3 %)	76.4 (2.5 %)	79.9 (+1.2 %)	79.3 (+2.5 %)	76.5 (+1.4 %)	85.2 (+5.4 %)	80.8 (+5.6)	75.3 (+0.5 %)
ComF13	86.9 (+5.0 %)	85.8 (+1.9 %)	83.0 (+4.5 %)	82.2 (+4.8 %)	77.1 (+1.1 %)	74.7 (+0.2 %)	82.1 (+4.0 %)	81.1 (+4.9 %)	75.2 (−0.2 %)	87.9 (+8.7 %)	81.9 (+7.0)	75.4 (+0.6 %)
ComF14	85.9 (+3.8 %)	87.1 (+3.4 %)	81.4 (+2.5 %)	81.1 (+3.4 %)	80.9 (+6.1 %)	73.9 (−0.8 %)	80.7 (+2.2 %)	79.9 (+3.3 %)	74.8 (−0.7 %)	83.8 (+3.7 %)	81.1 (+6.0)	75.2 (+0.4 %)
PromIGF12	88.1 (+6.5 %)	87.8 (+4.2 %)	86.8 (+9.3 %)	85.2 (+8.6 %)	85.1 (+11.6 %)	81.5 (+9.3 %)	86.8 (+10.0 %)	85.2 (+10.2 %)	82.1 (+8.8 %)	86.4 (+6.9 %)	85.6 (+11.8)	82.3 (+9.8 %)
PromIGF13	88.9 (+7.4 %)	88.2 (+4.7 %)	86.5 (+8.9 %)	87.0 (+10.9 %)	86.2 (+13.1 %)	82.3 (+10.4 %)	86.9 (+10.1 %)	87.1 (+12.6 %)	83.8 (+11.1 %)	88.2 (+9.1 %)	87.6 (+14.5)	80.7 (+7.7 %)
PromIGF14	87.8 (+6.1 %)	87.1 (+3.4 %)	85.7 (+7.9 %)	86.4 (+10.2 %)	83.2 (+9.1 %)	81.9 (+9.9 %)	85.0 (+7.7 %)	85.2 (+10.2 %)	82.7 (+9.6 %)	86.0 (+6.4 %)	86.1 (+12.5)	80.0 (+6.8 %)
PromMRMRF12	89.2 (+7.8 %)	88.8 (+5.4 %)	86.1 (+9.6 %)	86.0 (+9.6 %)	86.5 (+13.5 %)	82.8 (+11.1 %)	87.0 (+10.2 %)	86.3 (+11.6 %)	82.5 (+9.4 %)	87.6 (+8.4 %)	86.2 (+12.6)	81.3 (+9.7 %)
PromMRMRF13	**90.3** (+9.1 %)	**89.1** (+5.8 %)	87.1 (+9.3 %)	**87.5** (+11.6 %)	**87.9** (+15.3 %)	83.6 (12.2 %)	**88.3** (+11.9 %)	88.0 (+13.8 %)	84.1 (11.5 %)	**89.1** (+10.2 %)	**87.9** (+14.9)	82.2 (+8.5 %)
PromMRMRF14	88.7 (+7.2 %)	88.1 (+4.6 %)	85.5 (+7.6 %)	85.9 (+9.5 %)	84.8 (+11.2 %)	82.0 (10.0 %)	87.5 (+10.8 %)	**88.1** (+13.9 %)	83.6 (10.8 %)	87.2 (+7.9 %)	86.7 (+13.3)	80.4 (+7.3 %)

datasets. For example, ComF12 feature set produced F-measure of 87.0 % which is increased from 82.7 % for BMNB classifier on movie review dataset as shown in Table 3.9.

As feature vector lengths of all the composite features are huge in dimensions, it is required to filter the irrelevant and noisy features for better classification results. IG and mRMR feature selection methods are used for this purpose. Prominent composite features performed significantly better than the unigram features with very less feature sizes. For example, PromIG13 feature set produced F-measure of 88.9 % (+7.4 %) with 2244 features for movie review dataset using BMNB classifier as shown in Table 3.9. Similarly, all the composite features constructed using mRMR feature selection method performed better than corresponding composite features generated with IG. For example, PromIGF12 produced F-measure of 88.1 %, whereas PromMRMRF12 produced 89.2 % using BMNB classifier for movie review dataset as shown in Table 3.9. Experimental results showed that composite feature set PromMRMRF13 outperformed other feature sets. PromMRMR feature set produces the F-measure of 90.3 % (+9.1 %) with BMNB classifier on movie review dataset. It is due to the fact that by including the prominent bi-tagged feature, we only include important sentiment information unlike in the case of bigrams and dependency features, as bigrams contain a large number of noisy features and dependency feature suffers from overgeneralization problem. In addition, bi-tagged features include very less redundant features as compared to bigrams and dependency features.

mRMR feature selection method performs better than IG as IG selects relevant features based on reduction in uncertainty in identifying the class after knowing the value of the feature. It does not eliminate redundant features. In contrast, mRMR feature selection method discards redundant features which are highly correlated among features and retains relevant features having minimum correlation. It is intuitive that when unigram features are combined with other features like bigram, bi-tagged, or dependency features, redundancy among features would increase. Thus, it is clear that composite features contain more information but at the cost of redundancy. Redundancy can be reduced with the help of mRMR feature selection technique.

Proposed clustering features outperform all other features discussed in most of the cases in terms of performance with a very less number of feature vector length. For example, clustering features produce F-measure of 89.2 % (+7.8 %) with 801 feature vector length with BMNB classifier for movie review dataset as shown in Table 3.6. It is due to the fact that proposed clustering features alleviate the sparseness of the features. Clustering features solve the problem of unseen features while reducing the feature set by grouping of the features. Clustering features incorporate the semantic information by grouping of the features based on the semantic orientation values.

Proposed clustering features group the words with same sentiment score. For example, “great” and “good” can be used interchangeably, but convey same sentiment. In this way of feature representation, classifier will be based on the semantic score of the words and will not depend much on the words chosen by the author to express their opinion. Therefore, by grouping of the features on the basis of semantic orientation of words, classifier would be able to classify the document more accurately.

Clustering features are very efficient with respect to execution time as feature vector length reduces significantly. The main question with clustering features is to choose the number of clusters to be considered for machine learning algorithm. We empirically chose the total number of clusters to be taken. F-measure of clustering features with various cluster sizes is presented in Table 3.6. It is observed from the experiments that if cluster size is very small, then the performance is not well. It is due to the overgeneralization of the features with less number of clusters. In this case, most of the features are clustered together even if they are not having similar semantic orientation. Further, as we increase the cluster size, performance increases up to a certain limit and further, the performance degrades. It is due to the reason that as we increase the cluster size, grouping of the words in the cluster performs fine. But, as we further increase the cluster size, performance of the sentiment analysis model tends to perform similar to unigram sentiment analysis model. Experimental results show that proposed method performs better with cluster size that is equal to 10 % of the original feature vector length.

3.4.3.2 Performance of Feature Selection Technique

Information gain has been considered as best feature selection technique for sentiment analysis in the literature [1, 124]. Therefore, we compared the performance of mRMR feature selection technique with IG. It is clear from the experiments that both the feature selection techniques improve the performance considerably for all the feature sets with both the classifiers. mRMR feature selection technique performs better than IG. It is observed during experiments that mRMR and IG select approximately 65–70 % features in common for all the dataset considered. It is also observed that remaining 30–35 % features in IG feature set were those features which were correlated with other features. mRMR feature selection method is able to remove those redundant features that in turn enables to include more relevant features which IG method is unable to do. mRMR discards unwanted noisy features and retains only relevant feature with minimum correlation among features. That is why mRMR feature selection method performs better as compared to IG.

3.4.3.3 Performance of Machine Learning Algorithm

SVM classifier has been considered as best machine learning algorithm for sentiment analysis in the literature [1, 91, 124]. In addition, naive Bayes (NB) is also used frequently in the literature. Therefore, we compared the performance of BMNB classifier with SVM and NB classifiers; further, experimental results show that BMNB classifier performs better than SVM and NB classifiers for most of the cases as results shown in Tables 3.7, 3.8, and 3.9. For example, BMNB, SVM, and

NB classifiers give F-measure of 78.4 %, 76.2 %, and 74.2 %, respectively, using unigram features on book review dataset. Dependency among attributes inevitably decreases the power of NB classifier. mRMR selects the prominent features out of complete feature set, those that are not correlated among themselves. Experimental results showed that performance of BMNB increased significantly after eliminating the irrelevant and noisy features. For example, PromMRMRF1 feature set produced F-measure 88.1 %, 87.1 %, and 85.4 % with BMNB, SVM, and NB classifiers, respectively, on movie review dataset. This is due to the fact that prominent features extracted from mRMR feature selection technique are less likely to be dependent among themselves. BMNB after mRMR feature selection method performs better as compared to SVM in terms of accuracy and execution time because mRMR feature selection technique is capable of removing the correlation among the features.

3.5 Conclusions

In this chapter, we examined the performance of various popular feature extraction techniques, viz., unigrams, bigrams, and dependency features. We also proposed bi-tagged features for supervised sentiment analysis which contain sentiment-rich information with less noise. We also introduce a novel feature extraction method based on semantic clustering of features for sentiment analysis in this chapter. Proposed method groups the words into clusters to alleviate the data sparseness problem faced by machine learning methods.

We also introduced various composite and prominent features. Next, we examined that minimum redundancy maximum relevance (mRMR) feature selection technique performed better than state-of-the-art feature selection technique, i.e., IG. It is due to the fact that mRMR feature selection method is capable of selecting relevant features as well as it can eliminate redundant features unlike IG which can select only relevant feature. It is also investigated that both feature selection techniques improve the performance of the sentiment analysis. Further, it is observed that composite features produce improved results and composite feature set (PromMRMRF13) comprising of prominent unigrams (PromMRMRF1) and prominent bi-tagged (PromMRMRF3) features produced better results as compared to all other features. It is also investigated that BMNB classifier performed better as compared to state-of-the-art SVM classifier in most of the cases in terms of accuracy, and it is significantly better than SVM with respect to the execution time.

We also used trigrams as a feature vector for classification, but this feature vector performed worst for sentiment analysis. Performance of the trigram features was near to the random guessing of class, i.e., around 50 %. The main reason for this observation is the data sparseness of the trigram features, also due to the large number of noisy features; these results resemble with the finding of the Ng et al. [81]. Data sparseness of these features is intuitive as it is very rare to repeat the continuous three words in a testing document.

We combined each multi-word feature vector with the unigram features. As in our initial experiments, we found that other combinations of only multi-word features like bigrams, bi-tagged, and dependency features do not perform well due to the data sparseness. For example, combination of bigrams and dependency features does not perform well for sentiment analysis. Therefore, in this book, we create composite features by combining the multi-word features with the unigram features.

Chapter 4
Semantic Parsing Using Dependency Rules

Sentiment analysis from unstructured natural language text has recently received considerable attention from the research community. In the frame of biologically inspired machine learning approaches, finding good feature sets is particularly challenging yet very important. In this chapter, we focus on this fundamental issue of the sentiment analysis task. Specifically, we employ concepts as features and present a concept extraction algorithm to extract semantic features that exploit semantic relationships between words in natural language text. Additional conceptual information of a concept is obtained using the ConceptNet ontology. Concepts extracted from text are sent as queries to ConceptNet to extract their semantics. Further, we select important concepts and eliminate redundant concepts using the Minimum Redundancy and Maximum Relevance feature selection technique. All selected concepts are then used to build a machine learning model that classifies a given document as positive or negative.

4.1 ConceptNet

ConceptNet is a large semantic network consisting of large number of commonsense concepts [48, 65]. Commonsense knowledge in ConceptNet is contributed by ordinary people on the Internet. It is the largest machine usable commonsense resource consisting of more than 250,000 relations. It is the largest publicly available commonsense knowledge base which can be used to mine various inferences from the text. It consists of nodes (concepts) connected by edges (relations between concepts). Some of the relationships between concepts in the ConceptNet are IsA, EffectOf, CapableOf, MadeOf, DesireOf, etc. [48]. In ConceptNet, an assertion is defined by five properties, i.e., language, relation, concept1, concept2, and frequency. Here, concept1 and concept2 are the two concepts which are having a relation. Language property defines the language of the assertion (in our case

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_4

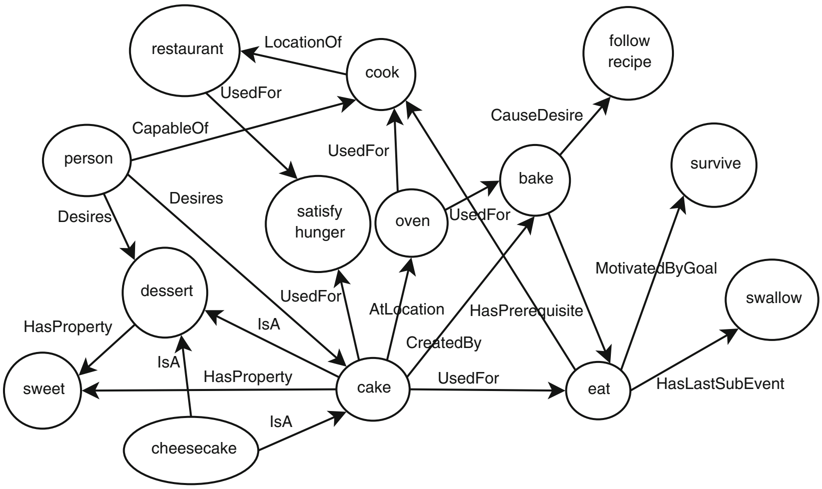

Fig. 4.1 Sample ConceptNet Ontology (This figure is adapted from Agarwal B., Mittal N., Bansal P., Garg S., "lSentiment Analysis Using Common-Sense and Context Information", In Computational Intelligence and Neuroscience, Article ID 715730, 9 pages, February 2015, DOI: http://dx.doi.org/10.1155/2015/715730, reprinted by the permission of the publisher (Hindawi Publishing Corporation, http://www.hindawi.com) under the Creative Commons Attribution License)

English). Frequency property represents how often given concepts are used with the given relation. Examples of ConceptNet relations, Restaurant UsedFor eat, Restaurant IsA place, etc.

The ConceptNet semantic graph represents the information from the Open Mind corpus as a directed graph, in which the nodes are concepts and the labeled edges are commonsense assertions that interconnect them. For example, given the two concepts "person" and "cook," an assertion between them is CapableOf, i.e., a person is capable of cooking as shown in Fig. 4.1 [48].

4.2 Syntactic N-Grams (sn-Gram)

Syntactic N-grams can be used as features for machine learning algorithm to learn the pattern of the text for sentiment analysis [115]. By dependency sn-gram, we understand a subtree of the dependency tree of a sentence that contains n nodes [113]. Syntactic N-grams can be used as features to represent sentences in the same scenarios as conventional n-grams; more specifically, sn-grams represent dependency trees as vectors in the same way as conventional n-grams represent strings of words. However, unlike conventional n-grams, sn-grams represent linguistic entities and are thus much more informative and less noisy. While sn-grams go a long way toward linguistically meaningful representation, numerous phenomena from the presence of functional words to synonymous expressions to insignificant

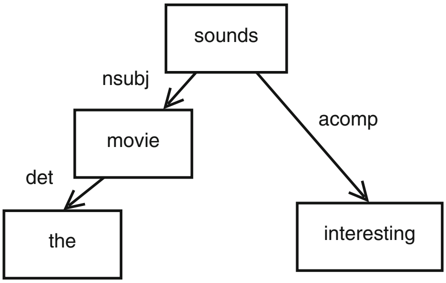

Fig. 4.2 Example: syntactic tree (This figure is adapted from Agarwal, B., Poria, S.,Mittal, N., Gelbukh, A., Hussain, A., "Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach", In Cognitive Computation, 20 January 2015, Volume 7, Issue 4, pp 487–499, reprinted with the permission of the publisher (Springer, http://link.springer.com/))

details still introduce noise in this representation and prevent semantically similar constructions to be mapped to identical feature vectors.

Syntactic N-grams contain the syntactic relations information between words in a sentence. In this feature extraction technique, words are taken by following syntactic relations in the sentence, not by taking the words as they appear in the text [114]. These features are extracted by following all the possible paths in the syntactic tree of the sentence to its leaf nodes. For example, in the sentence, "The movie sounds interesting," syntactic N-gram features are extracted as [sounds movie, sounds interesting, sounds movie the, sounds, movie, interesting]. Syntactic tree for the example sentence is shown in Fig. 4.2. These features bring syntactic information in the machine learning methods.

4.3 Proposed Methodology

Commonsense, in particular, is necessary to properly deconstruct natural language text into sentiments. For example, to appraise the concept of "small room" as negative for a hotel review and "small queue" as positive for a post office, or the concept "go read the book" as positive for a book review but negative for a movie review [21]. Commonsense knowledge describes basic understandings that people acquire through experience [51, 95, 102]. To this end, the presented concept parser aims to break text into clauses and, hence, deconstruct such clauses into concepts to be later fed to a vector space of commonsense knowledge. The presented concept parser is firstly published by Poria and his colleagues in the sourcePoria S., Agarwal B., Gelbukh A., Hussain A., Howard N., "Dependency-Based Semantic Parsing for Concept-Level Text Analysis", In Proceedings of the 15th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing), Vol: 8403, No: 1, 6 April 2014, pp. 113–127, reprinted with the permission of the publisher (Springer, http://link.springer.com/).

In the proposed approach, firstly, we extract dependency relations between the words of the sentence. Then, those relations are used to formulate complex concepts. Once, these concepts are extracted, we obtain related commonsense knowledge from ConceptNet. Further, only important concepts are selected and redundant concepts are eliminated using mRMR feature selection technique. Stanford dependency parser is used to retrieve dependency relations among words [70]. Below, we first describe the use of the dependency relations to form concepts, and later we discuss how related commonsense knowledge can be inferred from ConceptNet, and finally, mRMR feature selection technique is explained. Figure 4.3 demonstrates steps to be followed to evaluate the various proposed feature sets.

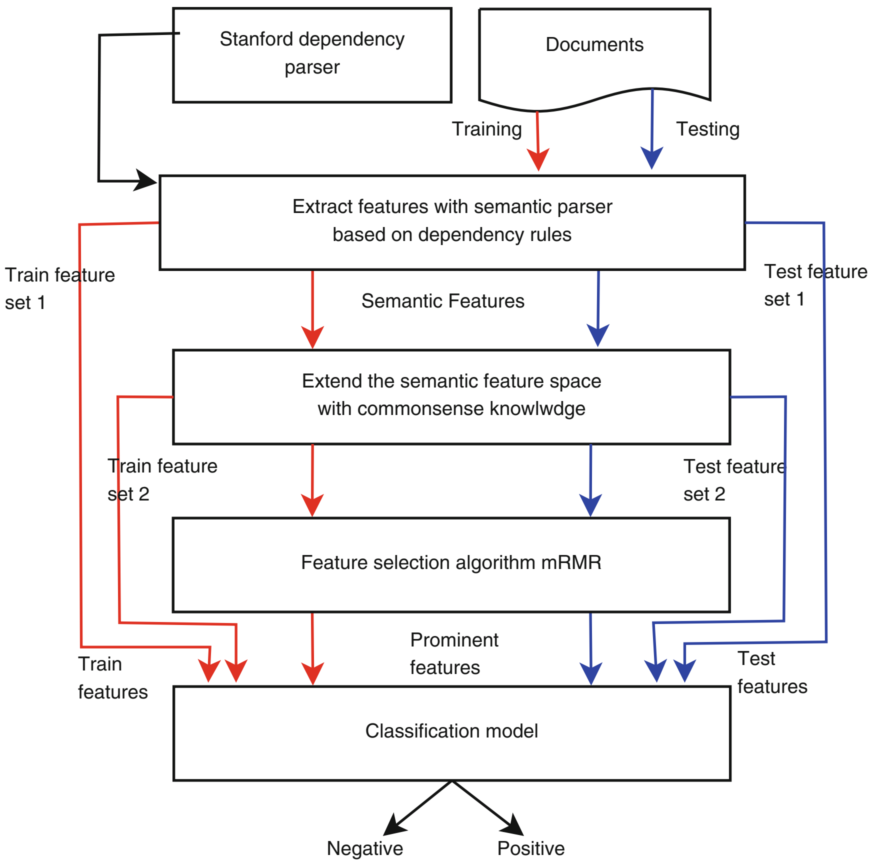

Fig. 4.3 Flow diagram of proposed approach for sentiment analysis (This figure is adapted from the source Agarwal, B., Poria, S.,Mittal, N., Gelbukh, A., Hussain, A., "Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach", In Cognitive Computation, 20 January 2015, Volume 7, Issue 4, pp 487–499, reprinted with the permission of the publisher (Springer, http://link.springer.com/))

4.3.1 Formation of Concepts Using Dependency Relations

Concepts are formed on the basis of the syntactic structures of the sentences. Dependency relation can be described as binary relation by the three types of elements.

1. Type
2. Head
3. Dependent

The type of the relation specifies the syntactic relation between two words in the sentence. The head of the relation is the pivot of the relation, and the main syntactic and semantic properties are inherited from the head. Dependent of the relation is the element that depends on the head word [96].

(i) Subject noun rule

Trigger: when the active token is found to be the syntactic subject of a verb.
Behavior: if a word h is in a subject noun relationship with a word t then the concept t-h is extracted.
Example: (1) The movie is boring. In (1), "movie" is in a subject relation with "boring." Here the concept (boring-movie) is extracted.

(ii) Joint subject noun and adjective complement rule

Trigger: when the active token is found to be the syntactic subject of a verb and the verb is on adjective complement relation with an adverb.
Behavior: if a word h is in a subject noun relationship with a word t and the word t is with adjective complement relationship with a word w then the concept w-h is extracted.
Example: (2) The movie sounds interesting. In (2), "movie" is in a subject relation with "sounds" and "sounds" is in adjective complement relationship with "interesting." Here the concept (interesting-movie) is extracted.

(iii) Direct nominal objects
This complex rule deals with direct nominal objects of a verb.

Trigger: when the active token is head verb of a direct object dependency relation.
Behavior: if a word h is in a direct nominal object relationship with a word t then the concept h-t is extracted.
Example: (3) Paul saw the movie in 3D.

In (3), the system extracts the concept (see, movie). (see, in, 3D) is not treated at this stage since it will later be treated by the standard rule for prepositional attachment.

(iv) Adjective and clausal complements rules

These rules deal with verbs having as complements either an adjective or a closed clause (i.e., a clause, usually finite, with its own subject).

Trigger: when the active token is head verb of one of the complement relations.
Behavior: if a word h is in a direct nominal object relationship with a word t, then the concept h-t is extracted.
Example: (4) The movie sounds boring.

In (4), "sounds" is the head of a clausal complement dependency relation with "boring" as the dependent. In this example, the concept (sound-boring) is extracted.

(v) Negation

Negation is also a crucial component of natural language text which usually flips the meaning of the text. This rule is used to identify whether a word is negated in the text.

Trigger: when in a text a word is negated.
Behavior: if a word h is negation by a negation marker t, then the concept t-h is extracted.
Example: (5) I do not like the movie.

In (5), "like" is the head of the negation dependency relation with "not" as the dependent. Here, "like" is negated by the negation marker "not."

Based on the rule described above the concept, (not, like) is extracted.

(vi) Open clausal complements

Open clausal complements are clausal complements of a verb that do not have their own subject, meaning that they (usually) share their subjects with that of the matrix clause. The corresponding rule is complex in the same way as the one for direct objects.

Trigger: when the active token is the head of the relation.
Behavior: as for the case of direct objects, the algorithm tries to determine the structure of the dependent of the head verb. Here the dependent is itself a verb; therefore, the system tries to establish whether the dependent verb has a direct object or a clausal complement of its own. In a nutshell, the system is dealing with three elements: the head verb (h), the dependent verb (d), and the (optional) complement of the dependent verb (t). Once these elements have all been identified, the concept (h,d,t) is extracted **Example**: (6) Paul likes to praise good movies.

In (6), "like" is the head of the open clausal complements dependency relation with praise as the dependent and the complement of the dependent verb praise is movie.

So, in this example the concept (like, praise, movie) is extracted.

(vii) Modifiers adjectival, adverbial, and participial modification

The rules for items modified by adjectives, adverbs, or participles all share the same format.

Trigger: these rules are activated when the active token is modified by an adjective, an adverb, or a participle.
Behavior: if a word w is modified by a word t, then the concept (t,w) is extracted.
Example: (7) a. Paul is a bad loser.

In (7), the concept (bad, loser) is extracted.

(viii) Prepositional phrases

Although prepositional phrases do not always act as modifiers, we introduce them in this section as the distinction does not really matter for their treatment.

Trigger: the rule is activated when the active token is recognized as typing a prepositional dependency relation. In this case, the head of the relation is the element to which the PP attaches, and the dependent is the head of the phrase embedded in the PP.
Behavior: instead of looking for the complex concept formed by the head and dependent of the relation, the system uses the preposition to build a ternary concept.
Example: (8) Bob hit Marie with a hammer.
In (8), the parser yields a dependency relation typed *prep_with* between the verb hit and the noun hammer (=the head of the phrase embedded in the PP).

Therefore, the system extracts the complex concept (hit, with, hammer).

(ix) Adverbial clause modifier

This kind of dependency concerns full clauses that act as modifiers of a verb. Standard examples involve temporal clauses and conditional structures.

Trigger: the rule is activated when the active token is a verb modified by an adverbial clause. The dependent is the head of the modifying clause.
Behavior: if a word t is an adverbial clause modifier of a word w, then the concept (t-w) is extracted.
Example: (9) The machine slows down when the best games are playing.

In (9), the complex concept (play, slow) is extracted.

(x) Noun compound modifier

Trigger: the rule is activated when it finds a noun composed with several nouns. A noun compound modifier of an NP is any noun that serves to modify the head noun.
Behavior: if a noun-word w is modified by another noun-word t, then the complex concept (t-h) is extracted.

Example: (10) Battery life of this phone is not good.

In (10), the complex concept (battery life) is extracted.

(xi) Single Word Concepts

Words having part of speech – verb, noun, adjective, and adverb – are also extracted from the text. Single word concepts which exist in the multi-word concepts are discarded as they carry redundant information, for example, concept "party" that already appears in the concept "birthday party," so we discard the concept "party."

4.3.2 Obtaining Commonsense Knowledge from ConceptNet

After obtaining concepts from the text, we send them as queries to ConceptNet. From ConceptNet, we find the commonsense knowledge related to the query concepts. For example, when we send the concept "birthday party" as a query to ConceptNet, we get related concepts such as "cake," "buy present." From ConceptNet, we find the following relations:

1. cake – AtLocation; birthday party.
2. buy present – UsedFor; birthday party.

These commonsense concepts are used to gather more knowledge about the concepts as they have direct connections with "birthday party." From ConceptNet, we get that cake is used in birthday party and people buy present for the birthday party. Thus, this process helps us to acquire more knowledge about the concepts we extract by the methodology described in previous section. Hence, the joint exploitation of the extracted concepts and ConceptNet offers to the machine a better understanding of the natural language text. Our approach enables the computer to understand the topic of the text as well as the meaning conveyed by the text.

4.3.3 Optimal Feature Set Construction

Feature set was constructed using semantic parsing scheme based on dependency rules, further appended using commonsense knowledge information. This feature set contains a lot of noisy and redundant information. Therefore, we use minimum redundancy maximum relevance (mRMR) feature selection technique to filter out irrelevant and noisy features to obtain optimal feature set. mRMR feature selection technique selects features which have high mutual information (maximum relevant) with the class attribute and eliminates features which have high mutual information (highly correlated) among themselves (minimum redundant). Detailed algorithm of mRMR technique is presented in Chap. 3 (Sect. 3.1.1).

4.4 Example

The difference between proposed feature extraction method and other state-of-the-art methods can be described with an example. Example sentence: “The movie sounds interesting.” In this example, unigrams are all the unique words. It is clear that unigram features are simple bag-of-words features which are unable to incorporate the semantic information. Bigrams features are able to extract some important information like “sounds interesting,” but it contains noisy features like “the movie,” and bi-tagged features contain useful information, but it loses a lot of important information. Dependency and syntactic features extract syntactic information considering the relations between the words in the sentence but with lots of noise. These features are also not able to include semantic information. In contrast, proposed semantic parser scheme is able to include more important and semantic feature, i.e., “interesting movie,” which is not extracted by any of the existing methods. Features extracted with various feature extraction methods for the example sentence “The movie sounds interesting” are presented in Table 4.1.

Another example sentence, “I can even now remember the hour from which I dedicated myself to this great enterprise.” In this example, unigrams features are [I, can, even, now, remember, the, hour, from, which, dedicated, myself, to, this, great, enterprise]. Bigram features are extracted by continuous sequence of two words as [I can, can even, even now, now remember, remember the, the hour, hour from, from which, which I, I dedicated, dedicated myself, myself to, to this, this great, great enterprise]. Next, dependency features extracted are as follows [remember I, remember can, now even, remember now, hour the, remember hour, dedicated which, dedicated I, remember dedicated, dedicated myself, enterprise this, enterprise great, dedicated enterprise]. Further, extracted syntactic n-grams are [remember now, now even, remember hour, remember dedicated, dedicated enterprise, enterprise great, remember now even, remember hour dedicated, hour dedicated enterprise, dedicated enterprise great]. Extracted concepts by our concept

Table 4.1 Example: comparison of various feature extraction methods (It is taken from the source Agarwal, B., Poria, S.,Mittal, N., Gelbukh, A., Hussain, A., “Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach”, In Cognitive Computation, 20 January 2015, Volume 7, Issue 4, pp 487–499, reprinted with the permission of the publisher (Springer, http://link.springer.com/))

Method	Features
Unigram	[The, movie, sounds, interesting]
Bigram	[The movie, movie sounds, sounds interesting]
Bi-tagged	[Sounds interesting]
Dependency features	[Sounds movie, sounds interesting, movie the]
Syntactic features	[Sounds movie, sounds interesting, sounds movie the, sounds, movie, interesting]
Proposed semantic parsing scheme	[Movie, movie sound, interesting movie, sound interesting]

parser are [even now, even now remember, remember hour, hour, remember from dedicate, dedicate which to enterprise, dedicate myself to enterprise, dedicate to enterprise, great enterprise].

After sending the extracted concepts as query to ConceptNet, in order to acquire more commonsense knowledge, we obtain the concept list [even now, even now remember, remember hour, hour, remember from dedicate, dedicate which to enterprise, dedicate myself to enterprise, dedicate to enterprise, great enterprise, still, sixty minute]. Here, we find commonsense knowledge from ConceptNet "still" and "sixty minute" related to the concepts "even now" and "hour," respectively. Clearly from above examples, we see that proposed concept parser is able to extract more semantic concepts. "Even now" and "even now remember" extracted by proposed concept parser express more semantic compared to "now even" and "remember now even" extracted by syntactic n-grams and other feature extraction techniques.

4.5 Results and Discussions

To evaluate the performance of the proposed methods, we used the same dataset as discussed in Chap. 3. F-measure is used as an evaluation metric in all the experiments. Details of datasets and evaluation metrics are presented in Chap. 3 (Sects. 3.4.1 and 3.4.2, respectively, for datasets used and evaluation metric). Tenfold cross-validation technique is used to evaluate the performance of the proposed methods; we randomly divide the dataset into 90 % training and 10 % testing documents, so that both the sets are disjoint. We construct the training model with 90 % of training dataset and further test the training model with 10 % of unused testing dataset. We repeat all the experiments 10 times, and the final performance is reported by averaging the results.

Feature extraction methods, namely, unigrams, bigrams, bi-tagged, dependency, clustering features, and various prominent and composite features, have been experimented as shown in Chap. 3. In continuation to these feature extraction methods, in this chapter we investigate the performance of two new feature extraction methods. First is the Syntactic N-grams feature set, which contains syntactic information among the words in a sentence. Syntactic N-grams are more informative and less arbitrary as compared to simple n-grams. In our knowledge, performance of sn-grams features has not been investigated for sentiment analysis. And the second is the proposed semantic concept extraction scheme, which extracts the semantic relationships within the words in the sentence using dependency parsing rules.

F-measure for various proposed feature sets with BMNB and SVM classifier on all the four datasets are reported in Table 4.2. Syntactic N-gram feature set produced the F-measure of 86.4 % with BMNB classifier for movie review dataset as shown in Table 4.2. This feature set performed well as compared to other simple feature extraction techniques like unigrams, bigrams, bi-tagged, and dependency features. Still, sn-gram features are not able to include semantic information. Next, proposed

Table 4.2 F-measure (in %) for various common-sense based features on four datasets (Partial results presented in this table are the published in the source Agarwal, B., Poria, S.,Mittal, N., Gelbukh, A., Hussain, A., "Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach", In Cognitive Computation, 20 January 2015, Volume 7, Issue 4, pp 487–499, reprinted with the permission of the publisher (Springer, http://link.springer.com/))

	Movie		Book		DVD		Electronics	
Feature extraction methods	BMNB	SVM	BMNB	SVM	BMNB	SVM	BMNB	SVM
Syntactic N-grams	86.4 (+4.4%)	87.3 (+3.6%)	84.3 (+7.5%)	84.6 (+11.0%)	84.0 (+6.4%)	84.1 (+8.7%)	86.4 (+6.9%)	85.8 (+12.1%)
With only Semantic parser	88.8 (+7.3 %)	86.4 (+2.6 %)	86.9 (+10.8 %)	86.1 (+12.9 %)	87.1 (+10.3 %)	86.8 (+12.2 %)	88.2 (+9.1 %)	87.1 (+13.8)
With Semantic parser and commonsense knowledge	89.7 (+8.4 %)	88.9 (+5.5 %)	86.0 (+9.6 %)	86.9 (+14.0 %)	86.9 (+10.1 %)	86.2 (+11.5 %)	87.3 (+8.0 %)	87.0 (+13.7)
With Semantic parser, commonsense knowledge and mRMR	**92.9 (+12.3 %)**	**90.1 (+7.0 %)**	**89.3 (+13.9 %)**	**88.5 (+16.1 %)**	**90.1 (+14.1 %)**	**89.2 (+15.3 %)**	**89.7 (+11.0 %)**	**88.9 (+16.2 %)**

semantic parser produced better results as compared to other feature sets explored so far. Semantic feature extraction method has produced F-measure of 88.8 % with BMNB classifier on movie review dataset as shown in Table 4.2. Semantic parser-based features with commonsense knowledge improved the results, giving F-measure of 89.7 % with BMNB classifier on movie review dataset as presented in Table 4.2. The main reason for this observation is that concepts extracted from ConceptNet include important information. But, the feature set constructed with concepts extracted from semantic parser in combination with commonsense knowledge does not contribute much as expected in sentiment analysis of review documents. It is due to the fact that ConceptNet-based commonsense knowledge contains a lot of noise in the form of irrelevant and redundant features together with important information. Therefore, when irrelevant and redundant features are eliminated from this feature vector using mRMR feature selection technique, it outperforms all other features.

For example, concept feature set produces F-measure of 92.9 % with BMNB classifier for movie review dataset as shown in Table 4.2. The main reason for this observation is that it extracts features which contain more useful sentiment information and also contain ConceptNet-based knowledge which was not present in previous feature sets. Another difference with previous methods is that syntactic n-grams convey all characteristics of basic n-gram, whereas the concept parser extracts semantic from the text. In addition, mRMR feature selection technique is also used which eliminates the redundant information and selects only important

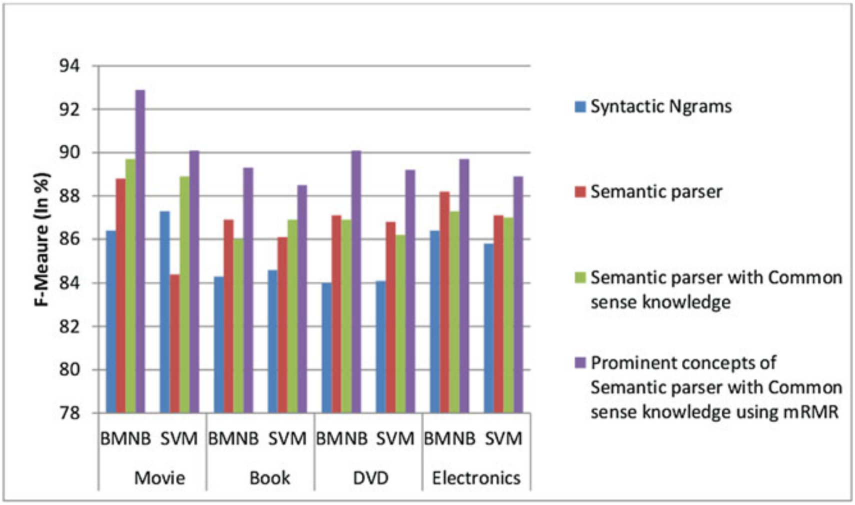

Fig. 4.4 Performance of proposed fetures with BMNB and SVM classifier with all the datasets

information for classification. Figure 4.4 shows the performance of all the features with BMNB and SVM classifier with all the datasets.

4.5.1 Comparison with Existing Methods

A performance comparison of our approach with state-of-the-art results reported in the literature is shown in Table 4.3. Researchers have mostly used the movie review and product review datasets to evaluate their methods for sentiment analysis. As can be seen from Table 4.3, we achieve results competitive with previous works on both the movie review and product review datasets. Semantic parsing scheme with commonsense knowledge and the mRMR feature selection method outperformed other methods reported in the literature on the same dataset. However, the results are not directly comparable due to different experimental settings used by the researchers with respect to evaluation measures, tools, and preprocessing methods.

For example, Matsumoto et al. [72], as part of the preprocessing step, employed unigram patterns that appear in at least two distinct sentences in the dataset. They reported 87 % accuracy for unigram features (simple bag-of-words features), whereas we report 84.2 % accuracy in our experiment for the same unigram feature set with a similar linear SVM classifier. In addition, they reported a performance accuracy of 85.1 % for bigram features, which is significantly better than the 78.8 % produced by our approach, and 77.1 % reported by Pang et al. [91] for the same bigram features.

Table 4.3 Comparison with existing methods (This table is taken from the source Agarwal, B., Poria, S.,Mittal, N., Gelbukh, A., Hussain, A., “Concept-Level Sentiment Analysis with Dependency-Based Semantic Parsing: A Novel Approach ”, In Cognitive Computation, 20 January 2015, Volume 7, Issue 4, pp 487–499, reprinted with the permission of the publisher (Springer, http://link.springer.com/))

Paper	Approach	Machine learning algorithm	Dataset	Best accuracy
Pang et al. [92]	Minimum cut algorithm	SVM	Movie review	87.1
Prabowo et al. [104]	Hybrid (rules+ closeness measure+ SVM)	SVM	Movie review	87.3
Mullen et al. [79]	Hybrid method with Turney and Osgood values	Hybrid SVM	Movie review	86
O’keefe et al. [86]	SentiWordNet based features and feature selection	SVM, NB	Movie review	87.15
Xia et al. [144]	Ensemble features	Ensemble classifier	Movie review, book, DVD, electronics	88.0, 83.0, 83.8, 86.0
Xia et al. [144]	Ensemble and generalized dependency features	SVM, NB	Movie reviews	88.6
Ng et al. [81]	Various dependency features	SVM	Movie review	90.5
Matsumoto et al. [72]	Various features such as unigrams, word subsequence and dependency subtree with frequent mining algorithm	SVM	Movie reviews	92.9
Tu et al. [131]	Dependency forest based features	MaxEnt	Movie review	91.6
Dang et al. [29]	Selected content free, content-specific and sentiment features	SVM	Book, DVD, electronics	78.85, 80.75, 83.75
Abbasi et al. [1]	Genetic Algorithms (GA), Information Gain (IG), IG + GA	SVM	Movie review	91.7
Abbasi et al. [2]	Stylistic and syntactic features	SVM	Movie reviews	87.9
Agarwal et al. [10]	Composite unigrams and bi-tagged feature with mRMR feature selection	BMNB	Movie review, book, DVD, electronics	90.3, 87.5, 88.3, 89.1
Proposed approach [12]	Dependency parsing based semantic parser with commonsense knowledge with mRMR	BMNB	Movie review, book, DVD, electronics	92.9, 89.3, 90.1, 89.7

The reason for this is the difference in experimental settings used by the respective researchers. For example, Matsumoto et al. considered only bigrams that occur at least twice in the dataset, in contrast to Pang et al. [91], who considered all bigrams that occur in at least seven documents. In addition, Matsumoto et al. [72] employed the Charniak parser to obtain dependency relations; while we used the Stanford dependency parser, Ng et al. [81] used the MINIPAR dependency parser. In addition, researchers use different machine learning tools to construct their models, for example, LibSVM, SVM^{light}, or WEKA.

Further, Matsumoto et al. [72] constructed their sentiment analysis model using various features, such as unigrams, word subsequence, and a dependency sub-tree. They also used the Frequent Pattern Mining software FREQT, available at http://www.chasen.org/~taku/software/, to mine frequent dependency tree patterns, and developed an SVM classifier by combining all of these features for sentiment classification. Their approach is computationally very expensive, as the feature vector length in their approach is huge due to the ensemble of variety of features. It is also computationally very expensive to apply the frequent pattern mining algorithm with dependency sub-tree features.

In contrast, we employed the mRMR feature selection technique to reduce the feature vector size; this resulted in a less computationally expensive procedure. In addition, Matsumoto et al. [72] reported that their method has the drawback of having overlap of features among word subsequence and dependency features. Feature set constructed with their method contains noisy and redundant information. On the other hand, the feature set constructed with our approach contains more important information due to the use of ConceptNet, while information redundancy is reduced through mRMR feature selection. Matsumoto et al. [72] reported a best performance of 92.9 %, which improves the baseline by +6.6 %, whereas our approach gives a best accuracy of 90.1 %.

Ng et al. [81] considered top-ranked features using weighted log-likelihood ratio (WLLR) feature selection from unigrams, bigrams, trigrams, and dependency features. They employed a manually constructed term polarity lexicon to select more polar bigrams and dependency features. They also used 2000 additional non-review documents to increase the list of objective features. Finally, they used the objective feature list to remove objective features from their original feature set. Their approach is rather complex, and the main drawback of their approach is that it requires human intervention. In contrast, our approach is fully automatic and significantly simpler. Ng et al. [81] reported a best performance of 90.5 %. which improves the baseline 87.1 % by +3.4%, while our approach gives a best accuracy of 90.1 %.

Xia et al. [144] experimented with various ensemble feature sets and ensemble classifiers to identify the best feature set and classifier for sentiment analysis on several datasets. Their approach faces the problem of noisy features due to the use of an ensemble of features. Xia and Zong [143] used various ensemble features and generalized dependency features for sentiment analysis. In particular, they focused on the problem of data sparseness by generalizing the dependency features. However, their approach is limited due to the problem of noisy features and overgeneralization of features.

Abbasi et al. [2] presented a sentiment analysis approach that filters irrelevant features with the help of a hybrid feature selection method. Their feature selection method is a combination of a genetic algorithm and information gain. A drawback of their approach is that it is highly computationally expensive to use a genetic algorithm for feature selection.

Tu et al. [131] proposed a technique to extract complex features using dependency forest in combination with unigrams, which is, again, computationally expensive. They employed tenfold cross-validation to evaluate the performance of their method with the MaxEnt classifier. We find that semantic parsing scheme with commonsense knowledge and the mRMR feature selection method outperforms other methods described in the literature on the same dataset.

4.6 Conclusions

Performance of sentiment analysis relies on the effectiveness of the feature extraction process. In this chapter, we presented a sentiment analysis approach based on machine learning. Initially, semantic feature extraction method is used to deconstruct the natural text that uses the dependency relations between words to extract the features from text. The joint exploitation of these concepts and ConceptNet helps to acquire more knowledge; thus, it enables a better understanding of the text. In addition, mRMR feature selection technique is used to select the important concepts and eliminate the redundant information.

Chapter 5
Sentiment Analysis Using ConceptNet Ontology and Context Information

Sentiment analysis research has been increasing tremendously in recent times due to the wide range of business and social applications. Sentiment analysis from unstructured natural language text has recently received considerable attention from the research community. In this chapter, we propose a novel sentiment analysis model based on commonsense knowledge extracted from ConceptNet-based ontology and context information. ConceptNet-based ontology is used to determine the domain-specific concepts which in turn produced the domain-specific important features. Further, the polarities of the extracted concepts are determined using the contextual polarity lexicon which we developed by considering the context information of a word. Finally, semantic orientations of domain-specific features of the review document are aggregated based on the importance of a feature with respect to the domain. The importance of the feature is determined by the depth of the feature in the ontology. Experimental results show the effectiveness of the proposed methods.

5.1 WordNet

WordNet is a large lexical database of English language consisting of synsets, i.e., cognitive synonyms. Synset expresses a distinct concept; it provides general definitions and provides short descriptions. WordNet has interlinking of concepts with conceptual-semantic and lexical relations. The resulting semantic network of related words and concepts can be used for inferring useful information for many natural language processing tasks. WordNet groups the words based on their meaning. It connects the synsets to other synsets with semantic relations. One of the important relations in WordNet is synonymy. For example, words "car" and "automobile" are semantically related. WordNet contains a total of 117,000 synsets which are linked to other synsets with semantic relations. In addition to this, synsets

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_5

Table 5.1 WordNet relations

Relation	Description
Hyperonymy, hyponymy, or ISA relation	It connects a synset to the more specific synset. For example, "bed" concept is linked to the more specific synset of "furniture"
Hyponymy	This relation is transitive: if an armchair is a kind of chair, and if a chair is a kind of furniture, then an armchair is a kind of furniture
Meronymy	Y is a meronym of X if Y is a part of X (window is a meronym of building)
Antonymy	Relation between terms with opposite meaning
Troponym	The verb Y is a troponym of the verb X if the activity Y is doing X in some manner (to lisp is a troponym of to talk)
Entailment	The verb Y is entailed by X if by doing X you must be doing Y (to sleep is entailed by to snore)

contain a brief definition (gloss) and most of the synsets illustrates the use of that synsets. All synsets are connected to other synsets by means of semantic relations. Some of these relations are demonstrated in Table 5.1.

5.2 Proposed Methodology

In the proposed approach [11], initially, we use ConceptNet to construct a domain-specific ontology for product reviews. Further, WordNet is used to expand the ontology for better coverage of the product features. Next, product features (aspects/entities) are identified using the ontology developed in the initial phase. We interchangeably use product features with aspects and entities. Some of the features from the camera domain are "battery life," "image quality," and "resolution." Similarly, "service," "ambience," "price," etc. are product features in restaurant domain. Next, sentiment expressing features/opinion features are extracted related to the product features, for example, "extremely comfortable," "good," "bad," "noisy," etc. Further, semantic orientations of these opinion features are determined with the help of contextual sentiment lexicon. Finally, overall sentiment orientation of the document is determined by aggregating the polarity values of all the opinion features corresponding to all the entities. Figure 5.1 presents the flow diagram of the proposed approach.

5.2.1 Construction of Domain-Specific Ontology from ConceptNet

An ontology can be described as connected concepts with semantic relationships. Semantic relations among concepts can be useful in inferring important informa-

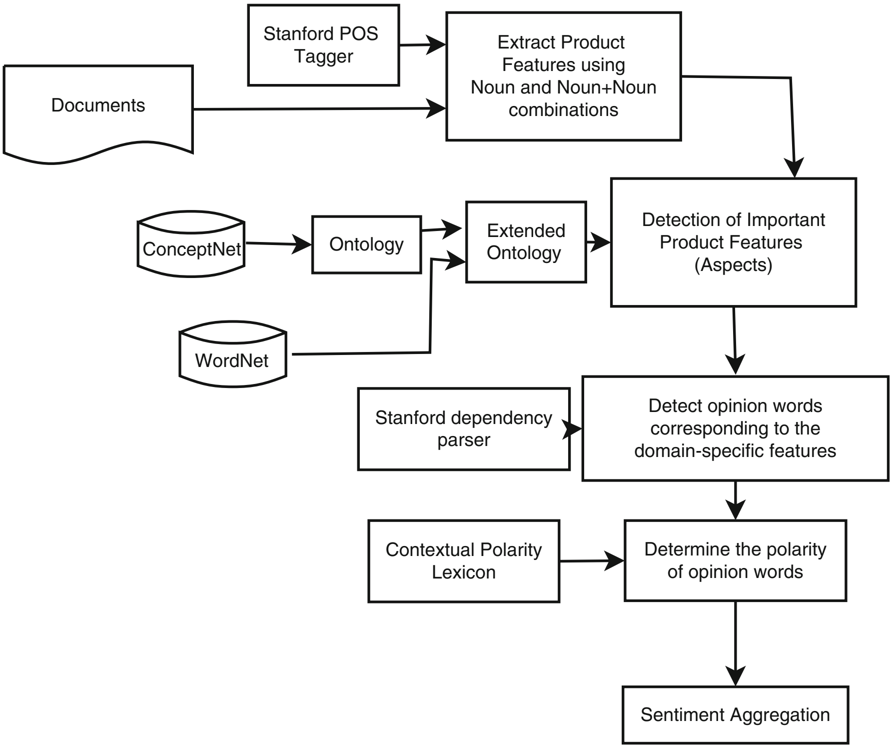

Fig. 5.1 Flow diagram of proposed approach based on common-sense and context information (This figure is adapted from the source Agarwal B., Mittal N., Bansal P., Garg S., "Sentiment Analysis Using Common-Sense and Context Information", In Computational Intelligence and Neuroscience, Article ID 715730, 9 pages, February 2015, DOI: http://dx.doi.org/10.1155/2015/715730, reprinted by the permission of the publisher (Hindawi Publishing Corporation, http://www.hindawi.com) under the Creative Commons Attribution License)

tion from the text. Constructed ontology can be considered as a commonsense knowledge base consisting of the domain-specific concepts and relations among them.

In the proposed approach, initially, we construct the domain-specific ontology with the help of ConceptNet. Next, we expand the ontology by merging the ontologies of the synonyms of the domain name for better coverage of domain-specific features. For example, to build the ontology for restaurant, we also construct ontology for "hotel" and further merge both the ontologies by connecting them with a new relation (i.e., IsEqualTo). The process of constructing the ontology is described in Algorithm 5.1. We extract the concepts from the ConceptNet up to level 4. The level of the ontology is set empirically. It is observed that as we increase the levels in ConceptNet, irrelevant concepts are extracted. Figure 5.2 demonstrates the sample ontology for restaurant domain.

Algorithm 5.1 Build domain-specific ontology from commonsense knowledgebase

INPUT Raw Assertions related to domain extracted from ConceptNet
OUTPUT Ontology with domain concepts

1: Every relation r in the ontology is constructed by connecting two concepts, i.e., concept1 (c1) and concept2 (c2).
2: Generate a graph structure using these relations. The root of this graph is the domain itself.
3: We connect two vertices V1 (i.e., concept1) and V2 (i.e., concept2) with an edge E (i.e., relation r). Connect all the nodes extracted from ConceptNet to construct the ontology.
4: First level nodes of this ontology are considered as new domain names and further synonyms are extracted from the WordNet for expansion of the ontology.
5: Repeat steps 1–3 to construct ontology for each synonym word of the main domain.
6: Merge all the extracted ontology to generate a single domain-specific ontology.

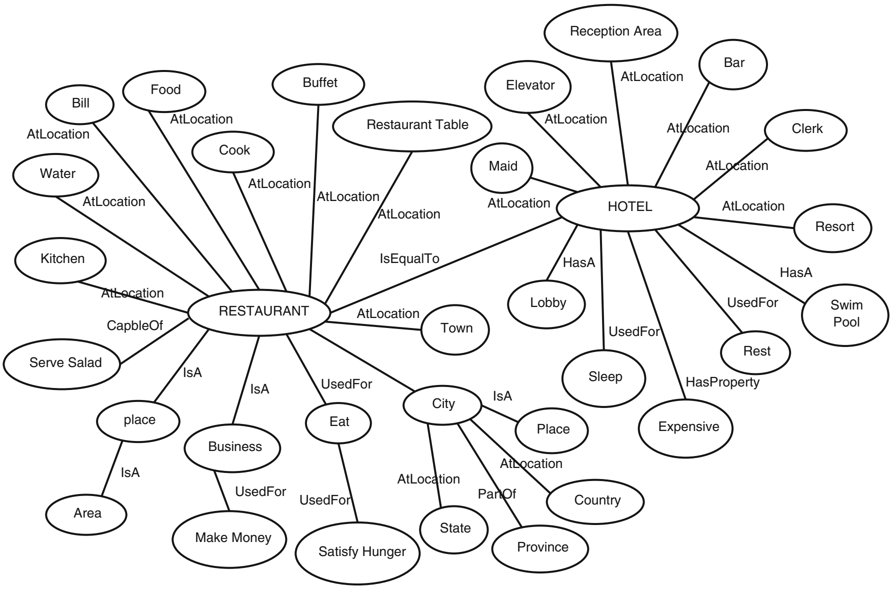

Fig. 5.2 Sample ontology for restaurant domain (This figure is adapted from the source Agarwal B., Mittal N., Bansal P., Garg S., "Sentiment Analysis Using Common-Sense and Context Information", In Computational Intelligence and Neuroscience, Article ID 715730, 9 pages, February 2015, DOI: http://dx.doi.org/10.1155/2015/715730, reprinted by the permission of the publisher (Hindawi Publishing Corporation, http://www.hindawi.com) under the Creative Commons Attribution License)

5.2.2 Aspect Extraction

Aspect is a term about which an opinion is expressed. In this step, we detect the aspects in a given sentence [43, 98]. For example, "this is a nice phone." Here, "phone" is an aspect about which sentiment is expressed. To identify the aspect term, first of all, reviews are part-of-speech tagged and further all the noun and noun + noun terms are extracted [52]. Stanford POS-tagger is used for tagging the review documents. For the same example, part-of-speech tagged sentence is "this_DT is_VBZ a_DT nice_JJ phone_NN." Here, "phone" is an aspect as it is noun. Another example is "it_PRP has_VBZ long_JJ battery_NN life_NN." Here, "battery life" is considered as an aspect due to noun + noun rule. After the extraction of noun and noun + noun-based aspects, these are matched with the domain-specific ontology constructed in Sect. 5.2.1 to eliminate all the unimportant and irrelevant aspects. All the irrelevant noun-based aspects are pruned with the help of ontology.

5.2.3 Feature-Specific Opinion Extraction

Users generally express opinion on multiple features of the product. The user may have a different opinion with respect to each feature in the review document. For example, "battery life of this phone is long, but appearance is bad." Here, opinion toward "battery life" is positive and negative toward "appearance." It is important to get the association of the opinion target (i.e., aspect) and opinion words [106, 148]. In this example, "battery life" is an aspect and "long" is an opinion word. To detect the association between opinion target and opinion words, dependency parsing is used [77]. Stanford dependency parser is used for extracting dependency rules from the review documents.

5.2.4 Construction of Contextual Polarity Lexicon

Sentiment lexicon is a dictionary containing the terms/words with their polarity value. We interchangeably use sentiment lexicon with polarity lexicon. We build the sentiment lexicon with the help of three publicly available resources, namely, SenticNet, SentiWordNet, and General Inquirer. Further, we determine the ambiguous terms from the sentiment lexicon and corresponding polarity depending upon the context in which terms appear. Finally, polarity of the opinion word is retrieved from sentiment lexicon and contextual polarity lexicon. The process of the construction of contextual sentiment lexicon is demonstrated in Fig. 5.3.

1. **SenticNet**
   SenticNet is a publicly available resource for sentiment analysis. It is a lexical resource constructed by clustering the vector space model of affective common-

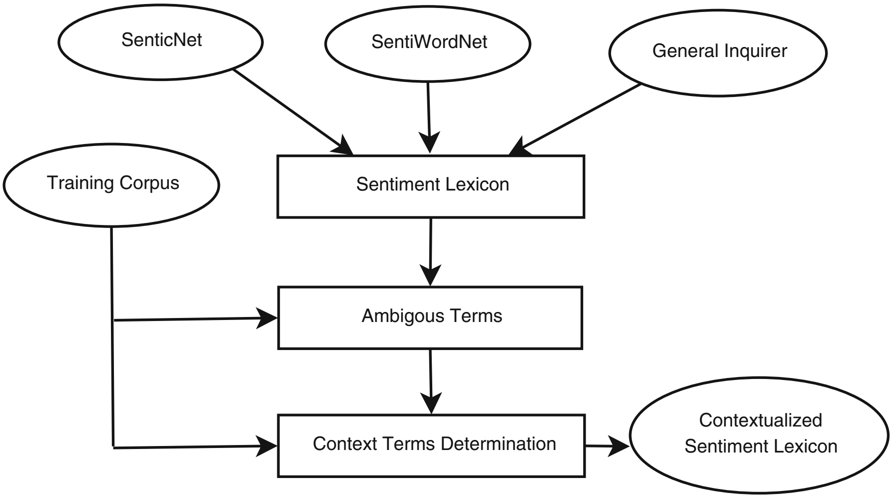

Fig. 5.3 Flow diagram of construction of contextual sentiment lexicon (This figure is adapted from the source Agarwal B., Mittal N., Bansal P., Garg S., "Sentiment Analysis Using Common-Sense and Context Information", In Computational Intelligence and Neuroscience, Article ID 715730, 9 pages, February 2015, DOI: http://dx.doi.org/10.1155/2015/715730, reprinted by the permission of the publisher (Hindawi Publishing Corporation, http://www.hindawi.com) under the Creative Commons Attribution License)

sense knowledge extracted from ConceptNet. This dictionary produces a list of concepts with their polarity value. Polarity value of the concepts given in this dictionary is computed by exploiting artificial intelligence (AI) and Semantic Web techniques. It contains more than 14,000 concepts along with their polarity scores. Polarity scores are in the range from −1.0 to +1.0. These concepts are a combination of single word and multiword concepts, having 7600 multiword concepts. Some of the concepts along with their polarity is shown in Table 5.2.

2. **SentiWordNet**
SentiWordNet is a sentiment dictionary containing polarity score of opinion words [37]. SentiWordnet contains polarity of the around two million words for adjectives, adverbs, verbs, and nouns, part-of-speech tagged words. Words in SentiWordNet are divided into four categories: adjective, adverb, verb, and noun. It can be obtained from (http://sentiwordnet.isti.cnr.it). SentiWordNet was built using WordNet and a ternary classifier. The classifier is based on "bag of synset" model which uses manually disambiguated glosses available from the Princeton WordNet Gloss Corpus. It is a WordNet-like lexicon which contains words with three scores as given below, i.e.,

   (i) Positive score
   (ii) Negative score
   (iii) Objective score

Table 5.2 Example concepts with their polarity in SenticNet

Concept	Polarity
Delicious fruit	+0.024
Baby cry	−0.627
Mindless entertainment	−0.055
Lose weight	+0.024
A lot	+0.970
Good character	+0.165
Mean person	−0.63
Seriously	+0.167
Investigation	−0.061
Express how funny	+0.541
Dead body	−0.129
Sad	−0.306
Negative feel	−0.877
Destroy	−0.106
Rush job	−0.321
Encouraged	+0.384
Pimple	−0.089
Say good bye	+0.046

Table 5.3 Example scores of some words in SentiWordNet

Words with senses	Positive	Negative	Objective
good#1	0.75	0	0.25
divine#1	0.875	0	0.125
solid#1	0.875	0	0.125
superb#2	0.875	0	0.125
abject#2	0	1	0
pitiful#2	0	1	0
bad#1	0	0.625	0.325
unfortunate#1	0	0.125	0.875

For every word, positive, negative, and neutral scores are having values between 0.0 and 1.0, and addition of all the scores, i.e., positive score, negative score, and objective score, for a word is 1. The objective score of 1.0 denotes that it is a neutral word and does not express any opinion. Some of the SentiWordNet scores are demonstrated in Table 5.3. There are a total of 147,306 English words in SentiWordNet.

3. **General Inquirer**

The General Inquirer (GI) was one of the first sentiment dictionaries. Publicly available dictionary is used for automatic text analysis [118]. It is a dictionary containing the 1195 positive and 2291 negative words. Some example words from positive and negative list are given in Table 5.4.

We combined all the three lexicons and prepared a sentiment lexicon which contains words with their polarity scores.

Table 5.4 Example words of General Inquirer

Positive words	Negative words
ADORNMENT	ABANDON
ADROIT	ABANDONMENT
ADULATION	ABATE
AFFILIATE	ABDICATE
AFFECTION	ABHOR
AFFINITY	ABJECT
AFFIRM	ABNORMAL
AFFLUENCE	ABOLISH
AFFLUENT	ABOMINABLE
AFLOAT	ABRUPT

5.2.4.1 Ambiguous Term Determination

The words which change their polarity depending on the context are called as ambiguous words [138]. For example, "I have a small complaint against this room; tap is not working properly in the bathroom." In this example, "complaint" is a term which has negative sentiment, but its polarity is changed to neutral due to the context term "small." The terms which occur dominantly in positive context and occur very less in negative context are most likely to be unambiguous terms. In contrast, the terms which occur equally in positive and negative contexts may be ambiguous terms. We use Algorithm 5.2 to detect such ambiguous terms from the sentiment lexicon.

Algorithm 5.2 Finding ambiguous terms

INPUT Sentiment lexicon and training corpus
OUTPUT Ambiguous terms, Pure Sentiment lexicon

```
1: for each Term (t_i) in Sentiment_Lexicon do
2:     μ_i = ((Positive_Count(t_i) * Positive_Score(t_i) − Negative_Count(t_i) * Negative_Score(t_i))/(Positive_Count(t_i) + Negative_Count(t_i))
3:     σ_i =((((μ_i − Positive_Score(t_i))^2 * Positive_Count(t_i)) − (μ_i − Negative_Score(t_i))^2 * Negative_Count(t_i)))/(Positive_Count(t_i) + Negative_Count(t_i)))^(1/2)
4: end for
5: if σ_i >= 0.75 then
6:     Ambiguity = YES
7:     Add to Ambigous_terms list
8: else
9:     Ambiguity = NO
10:    Add to Pure_Sentiment_Lexicon
11: end if
```

Initially, we compute the mean sentiment score (μ) and standard deviation score (σ) of all the sentiment terms from the sentiment lexicon with their given positive and negative sentiment score, along with their counts in positive and negative training documents. Mean sentiment score and standard deviation score are computed on the basis of distribution of occurrence of these terms in the training review documents. Further, the terms which have high standard deviation score are considered to be ambiguous terms. The threshold value for the mean standard deviation is determined empirically (demonstrated in Algorithm 5.2). All the terms which are not ambiguous are considered as pure polar/sentiment terms, i.e., they generally do not change polarity with the context.

5.2.4.2 Context Term Determination

Context terms are co-occurring terms with the ambiguous terms which can change the polarity of the term [141]. We need to consider the context term to determine the accurate polarity of the ambiguous terms. In this step, we find the context terms corresponding to each ambiguous term detected in previous step. And further these context terms are used to determine the polarity of the ambiguous term. Context terms are considered as all the nouns, adjectives, adverbs, and verbs that occurred two sentences before and after the occurrence of the ambiguous term in all the review documents [141]. Further, for each context term, we compute the probability that it belongs to positive sentiment class and negative sentiment class. Positive and negative class probabilities are computed as given in Eqs. (5.1) and (5.2).

$$\text{positive\_probability} = \frac{P}{(P+N)} \tag{5.1}$$

$$\text{negative\_probability} = \frac{N}{(P+N)} \tag{5.2}$$

Here, P is the total count of the term in positive training documents and N is the total count of the term in negative training documents. If the context term has significantly higher positive/negative class probability, it means that the term contributes more toward positive/negative polarity. Therefore, we consider a context term as positive/negative if it has a higher positive/negative probability. We sum up the polarities of all the context terms to determine the polarity of the ambiguous term. Finally, we compute the polarity for all the ambiguous terms determined in previous steps.

5.2.5 Sentiment Aggregation

1. First of all, features are extracted from the review document, then they are matched in the ontology. Importance of the feature is determined by the level of ontology where it is located. The features located at higher level near to root

of the ontology are considered to be more important as compared to lower-level features. Further, opinion word corresponding to this feature is detected using dependency parsing rules.

2. Further, the polarity of the opinion word is determined as follows. Initially, polarity of an opinion word is retrieved from contextual polarity lexicon, if the word is present. If the opinion word is not ambiguous, then polarity value is retrieved from pure sentiment lexicon (i.e., combination of SenticNet, SentiWordNet, and General Inquirer without ambiguous terms). Contribution of the opinion word in determining overall polarity is taken as polarity*(height of ontology).

   Finally, the overall polarity of the document is determined by summing up the contribution of each term.

5.3 Experiments and Results

5.3.1 Dataset and Evaluation

To evaluate the performance of the proposed methods, three standard benchmark datasets are used. First is the restaurant review dataset available at http://people.csail.mit.edu/bsnyder/naacl07/data/unformatted/. This corpus consists of 4,488 reviews. Reviews are classified as positive or negative polar documents. The second corpus is the movie review dataset provided by Pang and Lee [92]; it consists of 2000 reviews of equal number of negative and positive reviews. And the third dataset is the software review dataset, provided by Blitzer et al. [18]. It consists of 1000 positive and 915 negative review documents. Tenfold cross-validation method is used for evaluation of proposed methods. We randomly divide the dataset into 90 % training and 10 % testing documents, so that both the sets are disjoint. We repeat all the experiments 10 times, and final performance is reported by averaging the results. Accuracy is used as an evaluation measure. It is computed by dividing the total correctly classified testing documents by the total number of testing documents. All the features/words extracted from the review documents are lemmatized to reduce to their root form for better matching of features in the ontology.

5.3.2 Experiments

There are three objectives of this chapter: first is to investigate the effectiveness of the ontology used to prune the irrelevant features extracted from the review documents. It is done by selecting only domain-specific important features based on ConceptNet. Second is to investigate the effectiveness of incorporating the importance of the features in determining the overall sentiment of the document. Third is to evaluate the contextual polarity lexicon in determining the polarity of the opinion words.

5.3.2.1 Baseline

A simple lexical-based approach is considered as baseline in our experiments [77, 120, 137, 138]. In this approach, we use all the three sentiment lexicons, i.e., SenticNet, SentiWordNet, and General Inquirer (as discussed in Sect. 5.2.4) to retrieve the polarity value of all the features extracted from the review document. Then, we sum up the total positive and negative polarity values of all the words of the document; if total positive/negative polarity is greater than total negative/positive polarity value, then the document is assigned a positive/negative sentiment.

5.3.2.2 Domain-Specific Ontology-Based Method

In this experiment, we evaluate the contribution of the domain-specific ontology in improving the performance of sentiment analysis. In this experimental setting, we extracted the features with noun and (noun + noun) combinations from the review documents; further, extracted features are matched in the ontology to select only domain-specific important features. Then, dependency parsing is used to get the opinion words corresponding to the features extracted in the first phase. Then, all the three lexicons are used to get the polarity value of the opinion words. Finally, polarity values of all the opinion words in a document are summed to get the final polarity of the document.

5.3.2.3 Considering the Importance of the Features

This experiment is performed to investigate the effect of considering the importance of the feature in determining overall sentiment of the document. In this experiment, we consider the importance of the feature according to the depth of a feature in the domain specific ontology, the feature near the root node is considered more important to the domain and the feature matched at longer distance is considered to be less important.

5.3.2.4 Contextual Sentiment Lexicon

This experiment is to evaluate the performance of the sentiment analysis model when we consider the context information. In this experiment, we add the contextual polarity lexicon, in addition to all the three lexicons. The process to construct the contextual polarity lexicon is discussed in Sect. 5.2.4. All other settings are the same as method 2.

5.3.2.5 Considering Contextual Information and Importance of the Feature

In this experiment, we investigate the effect of the context information and the importance of the features and domain-specific ontology, all together to determine the efficiency of the proposed sentiment analysis model.

5.3.3 *Results*

Table 5.5 presents the results of all the experiments with three standard datasets. The baseline method gives the accuracy of 65.7 %, 67.8 %, and 70.1 % for restaurant review dataset. Next, the accuracy is improved with method 2 by incorporating domain-specific ontology to get only domain-related features as shown in Table 5.5. For example, accuracy is increased from 67.8 % to 69.2 % (+2.0 %) for the software review dataset. Further, method 3 improves the efficiency of the sentiment analysis model by considering importance of the features. Accuracy improves from 67.8 % to 72.6 % (+7.07 %) for software review dataset. Both methods are unsupervised; we only require a pre-defined ontology and pre-built polarity lexicons. It is observed from the experiments that the effect of commonsense knowledge-based ontology slightly depends on the type of the dataset. It improves the performance for the domains which have more possible aspects. For example, restaurant and software domain reviews have more possible aspects as compared to movie review dataset.

Further, performance is improved by considering the context information in determining the polarity of the opinion words for all the datasets (results as shown in Table 5.5). For example, accuracy is increased from 67.8 % to 77.3 % (+14.01 %) for software review dataset. Finally, performance is significantly increased from

Table 5.5 Accuracy (in %) of various methods on different datasets (This table is taken from the source Agarwal B., Mittal N., Bansal P., Garg S., "Sentiment Analysis Using Common-Sense and Context Information", In Computational Intelligence and Neuroscience, Article ID 715730, 9 pages, February 2015, DOI: http://dx.doi.org/10.1155/2015/715730, reprinted by the permission of the publisher (Hindawi Publishing Corporation, http://www.hindawi.com) under the Creative Commons Attribution License)

Method	Software	Movie	Restaurant
Baseline (Method 1)	67.8	70.1	65.7
Method 2 (With domain-specific ontology)	69.2 (+2.0 %)	71.3 (1.2 %)	68.3 (+3.9 %)
Method 3 (Considering importance of the feature)	72.6 (+7.07 %)	71.9 (+2.5 %)	71.1 (+8.2 %)
Method 4 (With contextual information)	77.3 (+14.01 %)	76.2 (+6.1 %)	76.2 (+15.9 %)
Method 5 (With context information and importance of the feature)	**80.1 (+18.14 %)**	**78.9 (+12.5 %)**	**79.4 (+20.8 %)**

67.8 % to 80.1 % (+18.14 %) for software domain when considering all the information, i.e., commonsense knowledge-based domain-specific ontology, importance of the extracted feature and contextual information.

5.3.4 Comparison with Related Work

The use of ConceptNet for sentiment analysis has not been explored much in the literature. Our proposed approach is quite similar to the approach proposed by Sureka et al. [119]. They developed domain-specific ontology from ConceptNet for target-specific sentiment analysis. Mukherjee and Joshi [77] proposed a method for sentiment aggregation using domain-specific ontology which was developed using ConceptNet. Our work is different from these approaches in the sense that we expanded the ontology with the help of WordNet for better coverage of the product features, and also we used contextual polarity lexicon developed by us to determine the polarity of the extracted features. Proposed approach produces best accuracy of 80.1 % on software review dataset, and Mukherjee and Joshi [77] gives best accuracy of 76.06 % on the same dataset.

5.4 Conclusions

In this chapter, we investigate the effect of three factors, i.e., domain-specific ontology, importance of the features with respect to the domain, and the contextual information all together in determining the overall sentiment of the text. The proposed approach including all these information significantly improves the performance of the sentiment analysis model over the baseline method.

Chapter 6
Semantic Orientation-Based Approach for Sentiment Analysis

Two types of techniques have been used in the literature for semantic orientation-based approach for sentiment analysis, viz., (i) corpus based and (ii) dictionary or lexicon or knowledge based. In this chapter, we explore the corpus-based semantic orientation approach for sentiment analysis. Corpus-based semantic orientation approach requires large dataset to detect the polarity of the terms and therefore the sentiment of the text. The main problem with this approach is that it relies on the polarity of the terms that have appeared in the training corpus since polarity is computed for the terms that are in the corpus. This approach has been explored well in the literature due to the simplicity of this approach [29, 120]. This approach initially mines sentiment-bearing terms from the unstructured text and further computes the polarity of the terms. Most of the sentiment-bearing terms are multi-word features unlike bag-of-words, e.g., "good movie," "nice cinematography," "nice actors," etc. Performance of semantic orientation-based approach has been limited in the literature due to inadequate coverage of the multi-word features.

In this chapter, we propose corpus-based semantic orientation approach for sentiment analysis. We investigate new sentiment-bearing feature extraction techniques in addition to traditional methods like unigrams, POS-based features, and dependency features. Further, a technique is presented which is useful to build the sentiment analysis model for the domains having only limited labeled dataset.

6.1 Proposed Approach

Proposed semantic orientation-based approach for sentiment analysis works as follows: Initially, various sentiment-bearing words/features are extracted. Further, semantic orientations of these sentiment-rich words or phrases are determined using

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_6

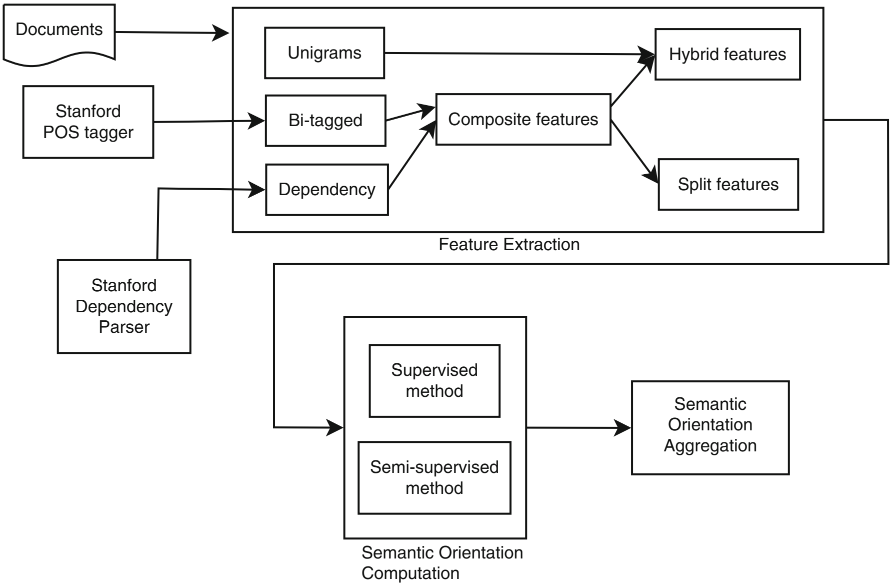

Fig. 6.1 Flow diagram of semantic orientation approach for sentiment analysis

various methods, and finally overall polarity of the document is determined by aggregating the semantic orientation of all the features [6]. Figure 6.1 demonstrates the flow diagram of the proposed approach.

6.1.1 Feature Extraction Methods

Semantic orientation-based approach relies on the sentiment-rich words. Various techniques like unigrams, bi-tagged, dependency composite, hybrid, and split features are explored to mine the sentiment-rich words and phrases from the text as discussed in the subsequent subsection.

6.1.1.1 Unigram Features

All the unique words in the corpus are considered as features if they conform to a specific part-of-speech (POS) tag, i.e., JJ (adjective), RB (adverb), NN (noun), and VB (verb). For example, in the sentence "*this_DT was_VB a_DT great_JJ movie_NN*." Here, the word "great" is an adjective and shows positive sentiment, while other words like "this," "was," and "a" are not conveying any sentiment in the text. Therefore, these words are dropped.

6.1.1.2 Bi-tagged Features

Bi-tagged phrases are extracted using part-of-speech (POS)-based fixed patterns. Bi-tagged phrases are selectively extracted which are intuitively sentiment-rich phrases. Technique used to extract the bi-tagged features is discussed in Sect. 3.3.1. Bi-tagged feature can include important contextual and sentiment information. For example, attaching an adverb like "very" with a polar adjective "good" will increase the polar intensity of the positive word "good" and contextual information like "not good," "unpredictable story," "amazing movie," etc. Therefore, two-word phrases are extracted that conform to the predefined pattern.

6.1.1.3 Dependency Features

A deeper linguistic analysis of syntactic relations between words in the sentence can be important for sentiment analysis. Wiebe and Riloff [140] investigated that syntactic patterns are very effective for subjective detection which is a prior step for sentiment analysis. In our experiments, we extract sentiment-bearing features conforming to the dependency relations presented in Table 6.1. Stanford dependency parser is used to generate the dependency relation for a given text. For example, all the dependency relations for the sentence "this movie is very nice" are "det(*movie_this*)," "nsubj(*nice_movie*)," "cop(*nice_is*)," and "advmod(*nice_very*)." Out of these dependency relations, "*nice_movie*," "*nice_is*," and "*nice_very*" features are selected according to the dependency rules given in Table 6.1.

6.1.1.4 Composite Features

New composite features are constructed by combining bi-tagged and dependency features as discussed in previous subsections. Bi-tagged feature extraction method is based on POS-tagged information; it is not able to extract all the sentiment-

Table 6.1 Selected dependency relations

S.no.	Relation	Meaning
1	acomp	Adjectival complement
2	advmod	Adverbial complement
3	amod	Adjectival modifier
4	dobj	Direct object
5	neg	Negation modifier
6	nsubj	Nominal subject
7	rcmod	Relative clause modifier
8	xcomp	Open clause complement
9	cop	Copula
10	ccomp	Clausal complement

rich phrases. Bi-tagged features can include contextual information but is unable to extract syntactic information as in case of dependency features which is also important for sentiment analysis. Therefore, by combining the two, we can extract more sentiment information in form of contextual and syntactic patterns of the text.

The main idea of composite features is to include more information by combining bi-tagged and dependency feature which can be explained by the following example: "This movie is very impressive and effective." POS-tagged sentence is as follows: "*This_DT Movie_NN is_VBZ very_RB impressive_JJ and_CC effective_JJ*." The feature "very impressive" would be extracted from this sentence using bi-tagged feature extraction technique. However, there is more sentiment-rich information which is intuitively very useful for the sentiment analysis that can be extracted using dependency features. Features extracted using dependency relations are as follows: nsubj(impressive, movie), nsubj(effective, movie), cop(impressive, is), advmod(impressive, very), and advmod(effective, very).

6.1.1.5 Hybrid Features

Multi-word features contain important sentiment information, but performance of most of the existing sentiment analysis models is limited due to the fact that these features are unable to contribute much because of less coverage in determining the overall sentiment of the text [6]. To improve the performance of the sentiment analysis, based on only multi-word features, we also include unigram information in the system. Hence, hybrid features are constructed by combining composite features and unigram features.

In hybrid features, sentiment information from both unigrams and multi-word features are taken into consideration for detection of overall semantic orientation of the document [17]. Contribution of unigrams and multi-word features is determined empirically in detecting the overall semantic orientation; it is observed that multi-word features are more important as compared to unigrams for sentiment analysis as it contains more sentiment information. Semantic score of a document is computed using Eq. (6.1):

$$SO(\text{doc}) = \left(\frac{3}{4}\right) * (\text{multi} - \text{word} - \text{feature}) + \left(\frac{1}{4}\right) * (\text{unigram}) \tag{6.1}$$

6.1.1.6 Split Features

Split features are constructed by dividing the multi-word features into single-word features. Split feature extraction method is also helpful in increasing the coverage of words in the corpus to determine the polarity of the document. Multi-word features match infrequently within other documents, sometimes none of the feature of testing document is previously seen in training document, and that document is difficult to classify in the positive or negative class [6]. The idea behind this technique is that

Table 6.2 Possible combination to compute polarity of multi-word features

Combinations	I word + polarity	II word + polarity	I word − polarity	II word − polarity	I word + polarity	II word − polarity	I word − polarity	II word + polarity
1	Avg		Avg		Avg		Avg	
2	Avg		Avg		Avg		Max	
3	Avg		Avg		Max		Max	
4	Avg		Max		Max		Max	
5	Max		Max		Max		Max	
6	Max		Max		Max		Avg	
7	Max		Max		Avg		Avg	
8	Max		Avg		Avg		Avg	
9	Avg		Avg		Max		Avg	
10	Avg		Max		Avg		Avg	
11	Max		Max		Avg		Max	
12	Avg		Avg		Avg		Avg	
13	Max		Avg		Max		Max	
14	Max		Avg		Avg		Max	

it is better to retrieve some information by splitting the feature, even if sometimes it disrupts the semantics of multi-word feature, rather than dropping the feature due to it being not present in the training corpus.

In this method, initially we extract the composite features. Next, we split the composite feature (multi-word feature) into single-word features if it is not present in the corpus (that multi-word feature doesn't occur in the training corpus). Next, semantic orientation of all the single words is computed separately, and further semantic orientation of the overall composite feature is determined by aggregating the semantic orientation of single words. Aggregation of semantic orientation is performed with all the possible combination as shown in Table 6.2, and the best combination is determined empirically.

To understand how aggregation is performed, consider the example combination 1 case I in Table 6.2, i.e., if the first word of the composite feature is having positive semantic orientation and the second word is having positive semantic orientation, then average polarity of the words is taken as semantic orientation of that composite feature.

6.2 Semantic Orientation

After extraction of various sentiment-bearing words, semantic orientations of these words are computed using supervised and semisupervised method as discussed in subsequent subsections.

6.2.1 Supervised Method

Computation of semantic orientation of the features is based on the assumption that if a feature is occurring frequently and predominantly in positive (negative) class, then that feature has high positive (negative) polarity. If a feature has high positive (negative) polarity value, then it indicates that the feature has high co-occurrence with positive (negative) words. Pointwise mutual information (PMI) is generally used to calculate the strength of association between a feature and positive or negative sentences. Equations to compute the semantic orientation of a feature is defined in Sect. 3.3.4.

6.2.2 Semisupervised Method

The problem with the supervised sentiment analysis model is the availability of limited labeled dataset for every domain. We propose a new semisupervised method requiring only a small number of labeled documents. In this method, initially we create a list of positive and negative seed words from the small number of labeled documents. Next, we compute the semantic orientations of all the features present in the document using mutual information based on the list of these positive and negative seed words.

The process to get the list of positive and negative seed words is as follows. Initially, unigram words/features are extracted, and further their positive and negative document frequencies are computed in labeled documents. Further, those words are selected as positive (negative) seed words if they satisfy the constraint c as given in Eq. (6.2). The assumption behind the constraint is that, more frequently occurring words in one class is strongly related to that class:

$$c = \begin{cases} \frac{(f_p+1)}{n*(f_n+1)} > 1,\ f_n \neq 0, & \text{for positive seed words} \\ \frac{(f_n+1)}{n*(f_p+1)} > 1,\ f_p \neq 0, & \text{for negative seed words} \end{cases} \tag{6.2}$$

Here, f_p is the frequency of a word in positive documents and f_n is the frequency of the word in negative documents. The value of n in the Eq. (6.2) is empirically determined to be 3. We apply this technique to obtain the seed words on randomly selected 100 positive and 100 negative documents. We obtained 40 positive and 34 negative seed words with the constraint given in Eq. (6.2). Examples of positive and negative seed words are given for the movie review dataset in Table 6.3. After construction of seed word list, semantic orientations of the features are computed using mutual information method. We used two methods to compute the semantic orientation of words using seed word list as discussed in subsequent sections.

Table 6.3 Sample list of words and their frequencies

Positive words			Negative words		
Word	Positive count	Negative count	Word word	Positive count	Negative count
Memorable	16	2	Worst	3	17
Effective	16	2	Boring	3	16
Powerful	17	4	Waste	2	16
Incredible	6	1	Terrible	3	14
Outstanding	7	1	Ridiculous	2	10
Wonderfully	8	1	Stupid	4	14
Fantastic	5	1	Lame	1	10

6.2.2.1 Mutual Information Method

Core idea of computation of semantic orientation of words with seed word list is quite similar to the supervised method that if a feature occurs frequently with positive (negative) seed words and also does not occur frequently with negative (positive) seed words, then that feature would have high positive (negative) polarity value. Semantic orientation of a feature c is computed using Eq. (6.3):

$$SO(c) = \log_2 \frac{p(c, \text{pos\_seed\_word})}{p(c, \text{neg\_seed\_word})} \tag{6.3}$$

$p(c, \text{pos\_seed\_word})$ is the probability of feature occurrence with positive seed words, and $p(c, \text{neg\_seed\_word})$ is the probability of feature occurrence with negative seed word in unlabelled document corpus. $p(c, \text{pos\_seed\_word})$ and $p(c, \text{neg\_seed\_word})$ are computed as shown in Eqs. (6.4) and (6.5):

$$p(c, \text{pos\_seed\_word}) = \frac{f(c, \text{pos\_seed\_word})}{f(\text{pos\_seed\_word})} \tag{6.4}$$

$$p(c, \text{neg\_seed\_word}) = \frac{f(c, \text{neg\_seed\_word})}{f(\text{neg\_seed\_word})} \tag{6.5}$$

$f(c, \text{pos\_seed\_word})$ is the number of documents containing both the feature c and positive seed word in the same document, and $f(\text{pos\_seed\_word})$ is the number of documents containing positive seed words. Similarly, $f(c, \text{neg\_seed\_word})$ is the number of documents containing both the feature c and negative seed word in the same document, and $f(\text{neg\_seed\_word})$ is the number of documents containing negative seed words.

6.2.2.2 Adaptive Mutual Information Method

Performance of mutual information method depends on the co-occurrence frequency of the features with the positive and negative seed words. Sometimes, semantic orientation of the feature is not computed correctly because both negative and positive seed words can occur in the same review; in addition, positive seed words can occur in negative reviews and vice versa. This can lead to inaccurate semantic orientation computation of features.

We propose an adaptive method for computation of semantic orientation based on the co-occurrence of a feature with positive or negative seed words. In this adaptive method, a sentence is initially labeled as positive or negative based on the number of positive and negative seed words present in that sentence. It is done by labeling the sentence positive if it has more positive seed words and otherwise labeled as negative. For example, considering a positive opinion, sentence “there is stupid acting by Mr. John but storyline of movie is wonderful and amazing direction” has positive seed words (wonderful, amazing) and a negative seed word (stupid). In this sentence, for a feature “amazing direction,” its co-occurrence frequency with positive seed word is 1 and negative seed word is also 1, resulting neutral polarity using the traditional method of polarity computation. Further, if adaptive mutual information method is used, initially the sentence is labeled positive as it has more positive seed words, so discarding co-occurrence frequency with negative seed words and only considering the co-occurrence frequency with positive seed words, resulting into overall positive polarity.

Considering another negative opinion, sentence “movie was ridiculous although nice cinematography but story was boring” as an example has a positive seed word (nice) and negative seed words (ridiculous, boring). In this sentence, for a feature “boring story,” its co-occurrence frequency with positive seed word is 1 and negative seed word is also 1, resulting neutral polarity using the traditional method of polarity computation. Further, with adaptive mutual information method, the sentence is labeled as negative, and its co-occurrence frequency with negative seed word is 1, discarding frequency with positive seed word, resulting into negative overall polarity of the feature.

6.2.3 Semantic Orientation Aggregation

After computation of semantic orientation of all the extracted features from the training documents, we have a lexicon of various features with their semantic orientation. Further, for the testing document, initially features are extracted and then semantic orientation values of these features are retrieved from the developed lexicon. Finally, overall polarity of the document is determined by summing up the semantic orientation of all the features in the document.

6.3 Experiment Result and Discussion

To evaluate the effectiveness of the proposed methods for sentiment analysis, two publicly available standard datasets are used. First dataset is movie review dataset also known as Cornell's Dataset (Pang and Lee [92]). Another dataset is product review dataset, provided by Amazon (Blitzer et al. [18]). It consists of various domain reviews. We used product reviews of books, DVDs, and electronics to evaluate the performance of all the proposed methods. Tenfold cross-validation technique is used to evaluate the performance of the proposed methods; we randomly divide the dataset into 90 % training and 10 % testing documents, such that both the sets are disjoint. We repeat all the experiments 10 times with randomly selected training and testing sets, and final performance is reported by average of the results. Accuracy is used as a performance evaluation measure which is computed by dividing total correctly classified documents by the total testing documents.

6.3.1 Results

We categorize the experiments as supervised and unsupervised on the basis of computation method used for semantic orientation of the features.

6.3.1.1 Supervised Method

Accuracies for various features with supervised method on various datasets are reported in Table 6.4. Unigram features do not perform well using semantic orientation-based approach; the main reason for this observation is that large corpus is required to compute the accurate polarity with PMI method. In all the experiments, we investigate the performance of all the proposed methods considering unigram method as baseline. Unigram features give an accuracy of 75.3 % on movie review dataset as shown in Table 6.4. Further, bi-tagged features are investigated for sentiment analysis. These features improve the accuracy from 75.3 % to 81.5 % (+8.2 %) on movie review as given in Table 6.4. It is due to the fact that bi-tagged features incorporate the contextual information which is very important for sentiment analysis. Further, dependency features are considered as features for sentiment analysis. These features further improve the accuracy up to 82.5 % (+9.5 %) for movie review as shown in Table 6.4. Dependency features produce better results as compared to unigrams and bi-tagged features due to more coverage and incorporation of syntactic information.

Further, performances of composite features are investigated, which gives an accuracy of 83.2 % (+10.4 %) for movie review dataset. These features improve the performance of sentiment analysis. It is due to the fact that these features provide

Table 6.4 Accuracy (in %) for various features with supervised method

Features	Movie	Book	DVD	Electronics
Unigrams	75.3	72.1	71.5	73.3
Bi-tagged	81.50 (+8.2 %)	75.0 (+4.0 %)	74.9 (+4.7 %)	74.0 (+0.9 %)
Dependency	82.5 (+9.5 %)	77.2 (+7.0 %)	76.2 (+6.5 %)	75.1 (+2.4 %)
Composite features	83.2 (+10.4 %)	78.3 (+8.5 %)	78.4 (+9.6 %)	77.8 (+6.1 %)
Hybrid features	84.3 (+11.9 %)	80.3 (+11.3 %)	79.1 (+10.6 %)	80.3 (+9.5 %)
Split features	**85.8 (+13.9 %)**	**81.2 (+12.6 %)**	**82.6 (+15.5 %)**	**82.9 (+13.0 %)**

Table 6.5 Accuracy for all the possible combinations for splitting features bi-tagged features on movie review dataset

Combinations	Correctly classified positive docs	Correctly classified negative docs	Correctly classified	Accuracy (in %)
1	278	207	485	81.83
2	278	217	495	83.50
3	274	218	492	83.00
4	**270**	**233**	**503**	**85.83**
5	278	205	483	81.00
6	279	199	478	80.00
7	280	194	474	80.00
8	283	181	464	78.33
9	275	210	485	80.83
10	274	222	496	82.67
11	270	228	498	83.00
12	280	203	483	81.50
13	281	195	476	80.33
14	284	187	471	79.50

more coverage of the features. Next, hybrid features are experimented for sentiment analysis; it incorporates the information of both unigrams and multi-word features.

Hybrid features give an accuracy of 84.3 % (+11.9 %) for movie review dataset. In further experiments, split features are investigated for sentiment analysis. In these features, all the possible methods of splitting multi-word features are empirically experimented, and results for all the combinations for bi-tagged features on movie review dataset are reported in Table 6.5. It is observed from the experiments that fourth combination produces best results. Therefore, results for splitting all the features with this combination are reported in Table 6.5. Split features produce better results as compared to other features. For example, split features produce the accuracy of 85.8 % (+13.9 %) on movie review dataset. The main reason for this is the increased coverage and inclusion of syntactic and contextual information.

Table 6.6 Accuracy (in %) for various features with semisupervised method

	Movie		Book		DVD		Electronics	
Features	MI	AMI	MI	AMI	MI	AMI	MI	AMI
Unigrams	69.6	72.8	68.6	70.1	67.9	70.1	68.1	70
Bi-tagged	72.0 (+3.4 %)	74.3 (+2.0 %)	71.1 (+3.6 %)	73.8 (+5.2 %)	68.3 (+0.5 %)	72.3 (+3.1 %)	69.3 (+1.7 %)	72.1 (+3.0 %)
Dependency	75.8 (+8.9 %)	78.5 (+7.8 %)	72.1 (+5.1 %)	75.1 (+7.1 %)	70.9 (+4.4 %)	74.2 (+5.8 %)	71.2 (+4.5 %)	74.4 (+6.2 %)
Composite features	77.4 (+11.2 %)	78.9 (+8.3 %)	73.9 (+7.7 %)	76.4 (+8.9 %)	73.2 (+7.8 %)	76.4 (+8.9 %)	73.9 (+8.5 %)	76.8 (+9.7 %)
Hybrid features	79.9 (+14.7 %)	82.0 (+12.6 %)	75.8 (+10.4 %)	77.8 (+10.9 %)	75.8 (+11.6 %)	79.9 (+13.9 %)	76.3 (+12.0 %)	79.2 (+13.1 %)
Split features	**81.6 (+17.2 %)**	**83.4 (+14.5 %)**	**76.3 (+11.2 %)**	**79.2 (+12.9 %)**	**77.7 (+14.4 %)**	**80.0 (+14.1 %)**	**78.9 (+15.8 %)**	**80.1 (+14.4 %)**

6.3.1.2 Semisupervised Method

Supervised methods face the problem of availability of labeled dataset for every domain. We propose a semisupervised method for sentiment analysis in which we just require a small number of labeled documents. Unigram features give an accuracy of 69.67 % on movie review dataset with mutual information technique used for polarity computation; it is considered as baseline method. This performance is further improved to 72.83 % (+4.5 %) by using proposed mutual information method for computing semantic orientation in place of traditional mutual information method as given in Table 6.6. Next, bi-tagged features produce accuracy of 72 % (+3.4 %) and 74.33 % (+2.0 %), respectively, for mutual information and adaptive mutual information-based methods with movie review dataset. Next, dependency features give best accuracy of 78.50 % (+7.8 %) for movie review dataset. Further, accuracy is improved with composite, hybrid, and split features similar to supervised methods due to the same reason. Performance of various features with both the methods of computing the semantic orientation, i.e., mutual information and adaptive mutual information on all the four datasets, is reported in Table 6.6.

Experimental results show that performances of supervised methods are superior to its corresponding semisupervised method. But it is also examined that semisupervised methods can produce good results and improve significantly over baseline method with efficient feature extraction technique and polarity computation method. Experimental results show that semisupervised methods can be very useful for the domains in which labeled dataset is a problem.

6.4 Conclusions

Efficiency of sentiment analysis model depends on the features which carry sentiment information of the text, and also, it is very important to assign accurate polarity values to these features for better performance. This chapter proposed various new feature extraction techniques for semantic orientation approach for sentiment analysis, which improve the performance significantly from baseline. Further, a new semisupervised approach is proposed for semantic orientation approach which is useful for the domains in which labeled dataset is a problem. Semisupervised methods are important because sentiment analysis is a domain-specific problem, and a model developed for one domain cannot perform well for other domains. Experimental results show that splitting of the multi-word features improves the performance of sentiment analysis for the domains having only limited labeled dataset.

Chapter 7
Conclusions and Future Work

The field of sentiment analysis is an exciting new research direction due to large number of real-world applications where discovering people's opinion is important in better decision-making. The development of techniques for the document-level sentiment analysis is one of the significant components of this area. Recently, people have started expressing their opinions on the Web that increased the need of analyzing the opinionated online content for various real-world applications. A lot of research is present in literature for detecting sentiment from the text. Still, there is a huge scope of improvement of these existing sentiment analysis models. Existing sentiment analysis models can be improved further with more semantic and commonsense knowledge.

In this chapter, we present conclusions drawn from our research work along with the possible directions of future work.

7.1 Conclusions

The aim of this book is to extract prominent features from the unstructured text that include semantic, syntactic, and commonsense knowledge for the document-level sentiment analysis. In this book, we employed both machine learning and semantic orientation-based approaches for sentiment analysis. We proposed new methods to improve the state-of-art methods in each category, and further we proposed a new sentiment analysis approach that include semantic concept parsing scheme and commonsense knowledge.

In this book, it is observed that machine learning methods perform better than other semantic orientation-based methods. However, machine learning methods require efficient feature extraction and feature selection technique for effective results. Semantic orientation-based sentiment analysis models perform well and are also useful when labeled training dataset is a problem.

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5_7

7.1.1 Summary of Main Findings

The main findings of this book are summarized as follows:

1. Experimental results show that linear combination of features improve the performance of the sentiment analysis. For example, composite features improve the performance of the sentiment analysis.
2. Performance of the sentiment analysis can be improved by reducing the redundancy among the features. In this book, experimental results show that minimum redundancy maximum relevance (mRMR) feature selection technique improves the performance of the sentiment analysis by eliminating the redundant features.
3. Boolean multinomial naive Bayes (BMNB) machine learning algorithm with mRMR feature selection technique performs better than support vector machine (SVM) classifier for sentiment analysis.
4. The problem of data sparseness is alleviated by semantic clustering of features. Semantic clustering of the features based on the similar semantic orientation values improves the performance of the sentiment analysis. Clustering features solves the problem of unseen features while reducing the feature set by grouping of the words.
5. Semantic relations among the words in the text have useful cues for sentiment analysis. Commonsense knowledge in form of ConceptNet ontology acquires knowledge, which provides better understanding of the text that improves the performance of the sentiment analysis.
6. Considering importance of the feature with respect to the domain improves the performance of the sentiment analysis. Domain-specific ontology can be used to determine the importance of the feature related to the domain, which is evident to be important for the sentiment analysis.
7. Splitting of the multi-word features improves the performance of sentiment analysis due to the more coverage of the features in the knowledge base for the domains in which labeled dataset is limitedly available.

7.1.2 Contributions

The main contributions of this book are summarized as follows:

1. We proposed to create various new composite features using unigram, bigram, and dependency parsing-based features which are able to incorporate information of single word, word sequence, and long-distance word dependency in the text. Combinations of various features improve the performance of the sentiment analysis. New POS-based bi-tagged features improve the performance of sentiment analysis. It produces F-measure of 86.9 (+5.0 %) for movie review dataset.
2. We were first to introduce minimum redundancy maximum relevance (mRMR) feature selection technique for sentiment analysis. Experimental results show that

minimum redundancy maximum relevance (mRMR) feature selection technique performs better than the state-of-art feature selection method IG.

3. In the literature, support vector machine (SVM) classifier is considered to be the best classifier for sentiment analysis. However, in this book, it is investigated that variant of naive Bayes classifier performs better than SVM classifier. Experimental results show that Boolean multinomial naive Bayes (BMNB) classifier performs better than SVM due to reduced redundancy features. Proposed hybrid method of mRMR and BMNB performs better as compared to other methods for sentiment analysis in terms of accuracy and execution time. It improves the performance up to (+9.1 %) from baseline for movie review dataset.
4. Proposed semantic clustering of features improves the performance of the sentiment analysis by grouping the similar semantic features. The proposed method alleviates the problem of data sparseness. Proposed clustering features improves the performance up to (+7.8 %) for movie review dataset.
5. Presented semantic feature extraction method uses the dependency relations between the words to extract the features from the text. The joint exploitation of these concepts and ConceptNet help to acquire more knowledge; thus, it enables a better understanding of the text. Experimental results show the effectiveness of the proposed semantic parsing scheme over other methods. Proposed sentiment analysis model gives F-measure of 92.9 %; it increases the performance by (+12.3 %) from the baseline unigram features.
6. We developed commonsense knowledge-based ontology for selecting exclusively domain-specific important features. Further, polarity computation technique is improved by combining multiple lexicons (like SentiWordNet, SenticNet, and General Inquirer) and contextual lexicon (to incorporate the context information) in determining the polarity value. Proposed sentiment analysis model improves the performance up to (+12.5 %) for movie review dataset.
7. Semantic orientation-based approach is improved with new proposed feature extraction methods, namely, composite, hybrid, and split features. Experimental results show that proposed feature extraction techniques significantly increase the performance of the semantic orientation-based approach for sentiment analysis. We also explore the semi-supervised approach for the domains in which labeled dataset is a problem. Proposed semi-supervised techniques with efficient features produce competitive results. Proposed split features with semi-supervised sentiment analysis model gives (+14.5 %) for movie review dataset.

7.2 Future Works

This field of research has been very attractive for the researchers and industrialists due to the large number of real-world applications. Further developments in the field of sentiment analysis are possible. We present some pointers to the possible future work as follows:

1. More dependency relations may be discovered to extract more semantic concepts.
2. Techniques to detect implicit opinion are still very limited; further new techniques may be explored for this problem.
3. New techniques may be explored to modify the proposed algorithms to be applied for non-English language.

Glossary

Feature Vector A feature vector is an n-dimensional vector of numerical features that represent some object, it is required to develop a machine learning model.

Composite Features Feature vector constructed by combinations of features.

ConceptNet It is a semantic network consisting of large number of common-sense concepts.

Prominent Features Feature vector constructed after eliminating irrelevant features.

Semantic Orientation Semantic orientation is the measure of the polarity strength i.e. positive or negative of a term/word.

Sentiment Analysis It is to determine the opinion expressed from the text.

Knowledge-base It is a kind of repository of information stored in a specific format.

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5

References

1. Abbasi A, Chen H, Salem A (2008) Sentiment analysis in multiple languages: feature selection for opinion classification in Web forums. ACM Trans Inf Syst 26(3):1–34
2. Abbasi A (2010) Intelligent feature selection for opinion classification. IEEE Intell Syst 25(4):75–79
3. Agarwal B, Mittal N (2012) Text classification using machine learning methods-a survey. In: Proceedings of the 2nd international conference on soft computing for problem solving (SocPros-2012), vol 236, no 1, Jaipur, pp 701–710
4. Agarwal B, Mittal N (2012) Categorical probability proportion difference (CPPD): a feature selection method for sentiment classification. In: Proceedings of the 2nd workshop on sentiment analysis where AI meets psychology, COLING 2012, Mumbai, pp 17–26
5. Agarwal B, Mittal N (2013) Sentiment classification using rough set based hybrid feature selection. In: Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis (WASSA'13), NAACL-HLT, Atlanta, pp 115–119
6. Agarwal B, Mittal N, Sharma VK (2013) Feature extraction methods for semantic orientation based approaches to sentiment analysis. In: Proceedings of the 10th international conference on natural language processing, Noida, pp 225–230
7. Agarwal B, Mittal N (2013) Optimal feature selection for sentiment analysis. In: Proceedings of the 14th international conference on intelligent text processing and computational linguistics (CICLing 2013), vol 7817, no 1, Samos, pp 13–24
8. Agarwal B, Mittal N, Cambria E (2013) Enhancing sentiment classification performance using bi-tagged phrases. In: Proceedings of the 13th IEEE international conference on data mining workshops, Dallas, pp 892–895
9. Agarwal B, Mittal N (2014) Semantic feature clustering for sentiment analysis of English reviews. IETE J Res Taylor Francis 60(6):414–422
10. Agarwal B, Mittal N (2014) Prominent feature extraction for review analysis: an empirical study. J Exp Theor Artif Intell. Taylor Francis. doi:10.1080/0952813X.2014.977830
11. Agarwal B, Mittal N, Bansal P, Garg S (2015) Sentiment analysis using common-sense and context information. Comput Intell Neurosci. Article ID 715730, 9. doi:http://dx.doi.org/10.1155/2015/715730
12. Agarwal B, Poria S, Mittal N, Gelbukh A, Hussain A (2015) Concept-level sentiment analysis with dependency-based semantic parsing: a novel approach. Cogn Comput 7(4):487–499
13. Agarwal B, Sharma VK, Mittal N (2013) Sentiment classification of review documents using phrases patterns. In: Second international symposium on natural language processing (NLP'13), Mysore, pp 1577–1580

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*,
Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5

14. Agarwal B, Mittal N, Sharma VK (2014) Semantic orientation based approaches for sentiment analysis. In: Issac B, Israr N (eds) Case studies in intelligent computing – achievements and trends. CRC, Taylor & Francis, pp 62–75
15. Agarwal B, Mittal N (2014) Machine learning approaches for sentiment analysis. In: Bhatnagar V (ed) Data mining and analysis in the engineering field. IGI Global, Hershey, pp 193–208
16. Aphinyanaphongs Y, Fu LD, Li Z, Peskin ER, Efstathiadis E, Aliferis CF, Statnikov A (2014) A comprehensive empirical comparison of modern supervised classification and feature selection methods for text categorization. J Assoc Inf Sci Technol 65(10):1964–1987
17. Bakliwal A, Arora P, Patil A, Verma V (2011) Towards enhanced opinion classification using NLP techniques. In: Proceedings of the 5th international joint conference on natural language processing (IJCNLP), Chiang Mai, pp 101–107
18. Blitzer J, Dredze M, Pereira F (2007) Biographies, bollywood, boom-boxes and blenders: domain adaptation for sentiment classification. In: Proceedings of the 45th annual meeting of association for computational linguistics (ACL), Prague, pp 440–447
19. Cambria E, Havasi C, Hussain A (2012) SenticNet 2: a semantic and affective resource for opinion mining and sentiment analysis. In: Proceedings of the 25th Florida artificial intelligence research society conference (FLAIRS), Marco Island, pp 202–207
20. Cambria E, Schuller B, Xia Y, Havasi C (2013) New avenues in opinion mining and sentiment analysis. IEEE Intell Syst 28(2):15–21
21. Cambria E, White B (2014) Jumping NLP curves: a review of natural language processing research. IEEE Comput Intell Mag 9(2):48–57
22. Cambria E, Fu J, Bisio F, Poria S (2015) AffectiveSpace 2: enabling affective intuition for concept-level sentiment analysis. In: Twenty-ninth AAAI conference on artificial intelligence, Austin, pp 508–514
23. Cambria E, Poria S, Bisio F, Bajpai R, Chaturvedi I (2015) The CLSA model: a novel framework for concept-level sentiment analysis. In: Computational linguistics and intelligent text processing, Cairo, pp 3–22
24. Cambria E, Poria S, Gelbukh A, Kwok K (2014) Sentic API: a common-sense based API for concept-level sentiment analysis. In: Proceedings of the 4th workshop on making sense of microposts (# Microposts2014), co-located with the 23rd international World Wide Web conference (WWW 2014), CEUR workshop proceedings, vol 1141, Seoul, pp 19–24
25. Chikersal P, Poria S, Cambria E (2015) SeNTU: sentiment analysis of tweets by combining a rule-based classifier with supervised learning. In: Proceedings of the international workshop on semantic evaluation, (SemEval 2015), Denver
26. Prerna C, Poria S, Cambria E, Gelbukh A, Siong CE (2015) Modelling public sentiment in Twitter: using linguistic patterns to enhance supervised learning. In: Computational linguistics and intelligent text processing. Springer International Publishing, Cham, pp 49–65
27. Cui H, Mittal V, Datar M (2006) Comparative experiments on sentiment classification for online product reviews. In: Proceedings of the 21st national conference on artificial intelligence, Boston, pp 1265–1270
28. Dai L, Chen H, Li X (2011) Improving sentiment classification using feature highlighting and feature bagging. In: Proceedings of the 11th IEEE international conference on data mining workshops (ICDMW), Vancouver, pp 61–66
29. Dang Y, Zhang Y, Chen H (2010) A lexicon enhanced method for sentiment classification: an experiment on online product reviews. IEEE Intell Syst 25(4):46–53
30. Dave K, Lawrence S, Pennock DM (2003) Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: Proceedings of the 12th international conference on World Wide Web (WWW), Budapest, pp 519–528
31. Denecke K (2008) Using SentiWordNet for multilingual sentiment analysis. In: Proceedings of the 24th international conference on data engineering workshop (ICDEW 2008), Cancun, pp 507–512

32. Deng ZH, Luo KH, Yu HL (2014) A study of supervised term weighting scheme for sentiment analysis. Expert Syst Appl 41(7):3506–3513
33. Devitt A, Ahmad K (2007) Sentiment polarity identification in financial news: a cohesion-based approach. In: Proceedings of the 45th annual meeting of the association of computational linguistics, Prague, pp 984–991
34. Dinu LP, Iuga I (2012) The Naive Bayes classifier in opinion mining: in search of the best feature set. In: Proceedings of the 13th international conference on intelligent text processing and computational linguistics, CICLing, vol 7181, no 1, New Delhi, pp 556–567
35. Duric A, Song F (2011) Feature selection for sentiment analysis based on content and syntax models. In: Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis, ACL-HLT, Portland, pp 96–103
36. Esuli A, Sebastiani F (2005) Determining the semantic orientation of terms through gloss analysis. In: Proceedings of the 14th ACM international conference on information and knowledge management (CIKM), Bremen, pp 617–624
37. Esuli A, Sebastiani F (2006) SentiWordNet: a publicly available lexical resource for opinion mining. In: Proceedings of 5th international conference on language resources and evaluation (LREC), Genoa, pp 417–422
38. Fahrni A, Klenner M (2008) Old wine or warm beer: target-specific sentiment analysis of adjectives. In: Proceedings of the AISB 2008 symposium on affective language in human and machine. The Society for the Study of Artificial Intelligence and Simulation of Behaviour Press, Aberdeen, pp 60–63
39. Fei Z, Liu J, Wu G (2004) Sentiment classification using phrase pattern. In: Proceedings of the fourth international conference on computer and infor-mation technology (CIT'04), Wuhan, pp 1147–1152
40. Ferreira L, Jakob N, Gurevych I (2008) A comparative study of feature extraction algorithms in customer reviews. In: Proceedings of the 2nd IEEE international conference on semantic computing, Santa Clara, pp 144–151
41. Forman G (2003) An extensive empirical study of feature selection metrics for text classification. J Mach Learn Res 3(1):1289–1305
42. Gamon M (2004) Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In: Proceedings of the 20th international conference on computational linguistics, Geneva, pp 841–848
43. Goujon B (2011) Text mining for opinion target detection. In: Proceedings of the European intelligence and security informatics conference (EISIC), Athens, pp 322–326
44. Guyon I, Elisseeff A (2003) An introduction to variable and feature selection. J Mach Learn Res 3(1):1157–1182
45. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. J SIGKDD Explor 11(1):10–18
46. Harlambous Y, Klyuev V (2013) Thematically reinforced explicit semantic analysis. Int J Comput Linguist Appl 4(1):79–94
47. Hatzivassiloglou V, McKeown KR (1997) Predicting the seman-tic orientation of adjectives. In: Proceedings of the thirty-fifth annual meeting of the Association for Computational Linguistics and the eighth conference of the European chapter of the Association for Computational Linguistics, Madrid, pp 174–181
48. Havasi C, Speer R, Alonso J (2007) Conceptnet 3: a flexible, multilingual semantic network for common sense knowledge. In: Proceedings of the international conference on recent advances in natural language processing (RANLP), pp 27–29
49. Hiroshi K, Tetsuya N, Hideo W (2004) Deeper sentiment analysis using machine translation technology. In: Proceedings of the 20th international conference on computational linguistics (COLING), Geneva, pp 494–500
50. Hoque N, Bhattacharyya DK, Kalita JK (2014) MIFS-ND: a mutual information-based feature selection method. Expert Syst Appl 41(14):6371–6385
51. Howard N, Cambria E (2013) Intention awareness: improving upon situation awareness in human-centric environments. Hum-centric Comput Inf Sci 3(9):1–17

52. Hu M, Liu B (2004) Mining and summarizing customer reviews. In: Proceedings of the 10th ACM international conference on knowledge discovery and data mining, Seattle, pp 168–177
53. Ikonomakis M, Kotsiantis S, Tampakas V (2005) Text classification using machine learning techniques. WSEAS Trans Comput 4(8):966–974
54. Joshi M, Penstein-Rose C (2009) Generalizing dependency features for opinion mining. In: Proceedings of the joint conference of the 47th annual meeting of the Association for Computational Linguistics (ACL), Singapore, pp 313–316
55. Kaji N, Kitsuregawa M (2007) Building lexicon for sentiment analysis from massive collection of HTML documents. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), Prague, pp 1075–1083
56. Kamps J, Marx M, Mokken RJ, Rijke MD (2004) Using wordnet to measure semantic orientation of adjectives. In: Proceedings of the 4th international conference on language resources and evaluation (LREC), Lisbon, pp 1115–1118
57. Kang H, Yoo SJ, Han D (2012) Senti-lexicon and improved Naive Bayes algorithms for sentiment analysis of restaurant reviews. Expert Syst Appl 39(5):6000–6010
58. Kennedy A, Inkpen D (2006) Sentiment classification of movie reviews using contextual valence shifters. Comput Intell 22(2):110–125
59. Kim SM, Hovy E (2004) Determining the sentiment of opinions. In: Proceedings of the 20th international conference on computational linguistics (COLING), Geneva, pp 1367–1373
60. Konig AC, Brill E (2006) Reducing human overhead in text categorization. In: Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining, Philadelphia, pp 598–603
61. Kumar A, Sebastian TM (2012) Sentiment analysis: a perspective on its past, present and future. Int J Intell Syst Appl 4(10):1–14
62. Li S, Zong C, Wang X (2007) Sentiment classification through combining classifiers with multiple feature sets. In: Proceedings of the international conference on natural language processing and knowledge engineering (NLP-KE), Beijing, pp 135–140
63. Lin Y, Zhang J, Wang X, Zhou A (2012) An information theoretic approach to sentiment polarity classification. In: Proceedings of the 2nd joint WICOW/AIRWeb workshop on web quality, Lyon, pp 35–40
64. Lin Y, Wang X, Zhang J, Zhou A (2012) Assembling the optimal sentiment classifiers. In: Proceedings of the 13th international conference on web information systems engineering, vol 7651, no 1, Paphos, pp 271–283
65. Liu H, Singh P (2004) ConceptNet – a practical commonsense reasoning tool-kit. BT Technol J Arch 22(4):211–226
66. Liu B (2010) Sentiment analysis and subjectivity. In: Indurkhya N, Damerau FJ (eds) Handbook of natural language processing, 2nd edn. Chapman & Hall/CRC, Boca Raton, pp 627–666
67. Liu B (2012) Sentiment analysis and opinion mining. Synthesis lectures on human language technologies. Morgan & Claypool Publishers, San Rafael
68. Maas AL, Daly RE, Pham PT, Huang D, Ng AY, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies, Portland, pp 142–150
69. Manning CD, Raghvan P, Schutze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
70. Marneffe D, Manning CD (2008) The Stanford typed dependencies representation. In: Proceedings of the workshop on cross-framework and cross-domain parser evaluation, Manchester. Association for Computational Linguistics, pp 1–8
71. Martineau J, Finin T (2009) Delta TFIDF: an improved feature space for sentiment analysis. In: Proceedings of the third AAAI international conference on weblogs and social media, pp 258–261
72. Matsumoto S, Takamura H, Okumura M (2005) Sentiment classification using word subsequences and dependency sub-trees. In: Proceedings of the 9th Pacific-Asia conference on advances in knowledge discovery and data mining (PAKDD), Hanoi, pp 301–311

73. Meena A, Prabhakar TV (2007) Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. In: Advances in information retrieval. Lecture notes in computer science, vol 4425, no 1, pp 573–580
74. Mejova Y, Srinivasan P (2011) Exploring feature definition and selection for sentiment classifiers. In: Proceedings of the fifth international AAAI conference on weblogs and social media, Barcelona, pp 546–549
75. Miller GA (1995) WordNet: a lexical database for English. Commun ACM 38(11):39–41
76. Moraes R, Valiati JF, Neto WPG (2013) Document-level sentiment classification: an empirical comparison between SVM and ANN. Expert Syst Appl 40(2):621–633
77. Mukherjee S, Joshi S (2013) Sentiment aggregation using ConceptNet ontology. In: Proceedings of the 6th international joint conference on natural language processing (IJCNLP), Nagoya, pp 570–578
78. Mukras R, Wiratunga N, Lothian R (2008) Selecting bi-tags for sentiment analysis of text. In: Proceedings of the 27th SGAI international conference on innovative techniques and applications of artificial intelligence, Cambridge, pp 181–194
79. Mullen T, Collier N (2004) Sentiment analysis using support vector machines with diverse information sources. In: Proceedings of the international conference on empirical methods in natural language processing (EMNLP), pp 412–418
80. Nakagawa T, Inui K, Kurohashi S (2010) Dependency tree-based sentiment classification using CRFs with hidden variables. In: Proceedings of the human language technologies: annual conference of the North American chapter of the Association for Computational Linguistics, Los Angeles, pp 786–794
81. Ng V, Dasgupta S, Arifin SMN (2006) Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In: Proceedings of the COLING/ACL 2006 main conference poster sessions, Sydney, pp 611–618
82. Nguyen DQ, Nguyen DQ, Pham SB (2013) A two-stage classifier for sentiment analysis. In: Proceedings of the 6th international joint conference on natural language processing, Nagoya, pp 897–901
83. Nguyen DQ, Nguyen DQ, Vu T, Pham SB (2014) Sentiment classification on polarity reviews: an empirical study using rating-based features. In: Proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis, Baltimore, pp 128–135
84. Nicholls C, Song F (2010) Comparison of feature selection methods for sentiment analysis. In: Proceedings of the 23rd Canadian conference on advances in artificial intelligence. LNCS, vol 6085, no 1, Ottawa, pp 286–289
85. Ohana B, Tierney B (2009) Sentiment classification of reviews using SentiWordNet. In: Proceedings of the 9th IT & T conference, Dublin, pp 1–9
86. O'keefe T, Koprinska I (2009) Feature selection and weighting methods in sentiment analysis. In: Proceedings of the 14th Australasian document computing symposium, Sydney, pp 67–74
87. Osajima I, Shimada K, Endo T (2005) Classification of evaluative sentences using sequential patterns. In: Proceedings of the 11th annual meeting of the association for natural language processing, Takamatsu, pp 1–8
88. Osgood CE, Succi GJ, Tannenbaum PH (1957) The measurement of meaning. University of Illinois Press, Urbana
89. Pak A, Paroubek P (2011) Text representation using dependency tree sub-graphs for sentiment analysis. In: Proceedings of the 16th international conference DASFAA workshop, vol 6637, no 1, Hong Kong, pp 323–332
90. Paltoglou G, Thelwallm M (2010) A study of information retrieval weighting schemes for sentiment analysis. In: Proceedings of the 48th annual meeting of the Association for Computational Linguistics, Uppsala, pp 1386–1395
91. Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP), Prague, pp 79–86

92. Pang B, Lee L (2004) A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the Association for Computational Linguistics (ACL), Barcelona, pp 271–278
93. Pang B, Lee L (2008) Opinion mining and sentiment analysis. Foundations and trends in information retrieval, vol 2, no 1–2. Now Publishers, Hanover, pp 1–135
94. Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 27(8):1226–1238
95. Poria S, Agarwal B, Gelbukh A, Hussain A, Howard N (2014) Dependency-based semantic parsing for concept-level text analysis. In: Proceedings of the 15th international conference on intelligent text processing and computational linguistics (CICLing), vol 8403, no 1, Kathmandu, pp 113–127
96. Poria S, Cambria E, Winterstein G, Huang G-B (2014) Sentic patterns: dependency-based rules for concept-level sentiment analysis. Knowl-Based Syst 69:45–63. http://dx.doi.org/10.1016/j.knosys.2014.05.005, ISSN 0950–7051
97. Poria S, Gelbukh A, Cambria E, Yang P, Hussain A, Durrani T (2012) Merging SenticNet and WordNet-affect emotion lists for sentiment analysis. In: 2012 IEEE 11th international conference on signal processing (ICSP), 21–25 Oct 2012, vol 2, Beijing, pp 1251–1255
98. Poria S, Cambria E, Ku L-W, Gui C, Gelbukh A (2014) A rule-based approach to aspect extraction from product reviews. SocialNLP 2014:28
99. Poria S, Gelbukh A, Cambria E, Das D, Bandyopadhyay S (2012) Enriching SenticNet polarity scores through semi-supervised fuzzy clustering. In: 2012 IEEE 12th international conference on data mining workshops (ICDMW), Brussels, pp 709–716
100. Poria S, Gelbukh A, Cambria E, Hussain A, Huang G-B (2014) EmoSenticSpace: a novel framework for affective common-sense reasoning. Knowl-Based Syst 69:108–123
101. Poria S, Gelbukh A, Hussain A, Howard N, Das D, Bandyopadhyay S (2013) Enhanced SenticNet with affective labels for concept-based opinion mining. IEEE Intell Syst 28(2):31–38. doi:10.1109/MIS.2013.4
102. Poria S, Gelbukh A, Agarwal B, Cambria E, Howard N (2013) Common sense knowledge based personality recognition from text. In: Castro F, Gelbukh A, González M (eds) Advances in soft computing and its applications. Springer, Heidelberg, pp 484–496
103. Poria S, Cambria E, Howard N, Huang G-B, Hussain A (2015) Fusing audio, visual and textual clues for sentiment analysis from multimodal content. Neurocomputing 174:50–59. Part A, Published 22 January 2016. http://dx.doi.org/10.1016/j.neucom.2015.01.095
104. Prabowo R, Thelwall M (2009) Sentiment analysis: a combined approach. J Informetr 3(2):143–157
105. Qiu G, Liu B, Bu J, Chen C (2009) Expanding domain sentiment lexicon through double propagation. In: Proceedings of the 21st international joint conference on artificial intelligence (IJCAI), pp 1199–1204
106. Qiu G, Liu B, Bu J, Chen C (2011) Opinion word expansion and target extraction through double propagation. J Comput Linguist 37(1):9–27
107. Raychev V, Nakov P (2009) Language-independent sentiment analysis using subjectivity and positional information. In: Proceedings of the international conference recent advances on natural language processing (RANLP), Borovets, pp 360–364
108. Read J (2005) Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In: Proceedings of the ACL student research workshop, Michigan, pp 43–48
109. Riloff E, Patwardhan S, Wiebe J (2006) Feature subsumption for opinion analysis. In: Proceedings of the conference on empirical methods in natural language processing, Sydney, pp 440–448
110. Saeys Y, Inza I, Larranaga P (2007) A review of feature selection techniques in bioinformatics. J Bioinform 23(19):2507–2517
111. Saleh MR, Martin-Valdivia MT, Montejo-Raez A, Urena-Lopez LA (2011) Experiments with SVM to classify opinions in different domains. Expert Syst Appl 38(12):14799–14804

112. Salvetti F, Lewis S, Reichenbach C (2004) Automatic opinion polarity classification of movie reviews. Colo Res Linguist 17(1):1–15
113. Sidorov G, Velasquez F, Stamatatos E, Gelbukh A, Chanona-Hernandez L (2013) Syntactic dependency-based N-grams as classification features. In: Proceedings of the Mexican international conference on artificial intelligence (MICAI), vol 7630, no 1, pp 1–11
114. Sidorov G (2013) Syntactic dependency based n-grams in rule based automatic English as second language grammar correction. Int J Comput Linguist Appl 4(2):169–188
115. Sidorov G, Velasquez F, Stamatatos E, Gelbukh A, Chanona-Hernandez L (2014) Syntactic n-grams as machine learning features for natural language processing. Expert Syst Appl 41(3):853–860
116. Simeon M, Hilderman R (2008) Categorical proportional difference: a feature selection method for text categorization. In: Proceedings of the 7th Australasian data mining conference, Glenelg, pp 201–208
117. Singhal K, Agarwal B, Mittal N (2015) Modeling Indian general elections: sentiment analysis of political Twitter data. In: Second international conference on information systems design and intelligent applications, Vol 339, pp 469–477
118. Stone PJ, Hunt EB (1963) A computer approach to content analysis: studies using the general inquirer system. In: Proceedings of the AFIPS, Detroit, pp 241–256
119. Sureka A, Goyal V, Correa D, Mondal A (2010) Generating domain-specific ontology from common-sense semantic network for target specific sentiment analysis. In: Proceedings of the fifth international conference of the Global WordNet Association, Mumbai, pp 1–8
120. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Comput Linguist 37(2):267–307
121. Takamura H, Inui T, Okumura M (2005) Extracting semantic orientations of words using spin model. In: Proceedings of the Association for Computational Linguistics (ACL), pp 133–140
122. Takamura H, Inui T, Okumura M (2006) Latent variable models for semantic orientations of phrases. In: Proceedings of the 11th European chapter of the Association for Computational Linguistics (EACL), pp 201–208
123. Takamura H, Inui T, Okumura M (2007) Extracting semantic orientations of phrases from dictionary. In: Proceedings of the joint human language technology/North American chapter of the ACL conference (HLT-NAACL), pp 292–299
124. Tan S, Zhang J (2008) An empirical study of sentiment analysis for chinese documents. Expert Syst Appl 34(4):2622–2629
125. Tan LKW, Na JC, Theng YL, Chang KY (2011) Sentence-level sentiment polarity classification using a linguistic approach. In: Proceedings of the 13th international conference on Asia-Pacific digital libraries (ICADL), Beijing, pp 77–87
126. Tan LKW, Na JC, Theng YL, Chang KY (2012) Phrase-level sentiment polarity classification using rule based typed dependencies and additional complex phrases consideration. J Comput Sci Technol 27(3):650–666
127. Tang H, Tan S, Cheng X (2009) A survey on sentiment detection of reviews. Expert Syst Appl 36(7):10760–10773
128. Thet TT, Na JC, Khoo CSG, Shakthikumar S (2009) Sentiment analysis of movie reviews on discus-sion boards using a linguistic approach. In: Proceedings of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion, Hong Kong, pp 81–84
129. Tsutsumi K, Shimada K, Endo T (2007) Movie review classification based on a multiple classifier. In: Proceedings of the annual meetings of the Pacific Asia conference on language, information and computation (PACLIC), pp 481–488
130. Tsytsarau M, Palpanas T (2012) Survey on mining subjective data on the web. Data Min Knowl Discov 24(3):478–514
131. Tu Z, Jiang W, Liu Q, Lin S (2012) Dependency forest for sentiment analysis. In: Proceedings of the first CCF conference on natural language processing and Chinese computing, Beijing, pp 69–77

132. Turney P (2002) Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp 417–424
133. Turney P, Littman ML (2003) Measuring praise and criticism: inference of semantic orientation from association. ACM Trans Inf Syst 21(4):315–346
134. Verma S, Bhattacharyya P (2009) Incorporating semantic knowledge for sentiment analysis. In: Proceedings of the international conference on natural language processing (ICON), Hyderabad, pp 1–6
135. Wang S, Li D, Song S, Wei Y, Li H (2009) A feature selection method based on Fisher's discriminant ratio for text sentiment classification. In: Proceedings of the international conference on web information systems and mining (WISM), Shanghai, pp 88–97
136. Wang L, Wan Y (2011) Sentiment classification of documents based on latent semantic analysis. In: Proceedings of the international conference on advanced research on computer education, simulation and modeling (CESM), Wuhan, pp 356–361
137. Wei CP, Chen YM, Yang CS, Yang CY (2010) Understanding what concerns consumers: a semantic approach to product feature extraction from consumer reviews. Inf Syst E-Bus Manag 8(2):149–167
138. Weichselbraun A, Gindl S, Scharl A (2013) Extracting and grounding context-aware sentiment lexicons. IEEE Intell Syst 28(2):39–46
139. Whitelaw C, Garg N, Argamon S (2005) Using appraisal groups for sentiment analysis. In: Proceedings of the 14th ACM international conference on information and knowledge management, Bremen, pp 625–631
140. Wiebe J, Riloff E (2005) Creating subjective and objective sentence classifiers from unannotated texts. In: Proceedings of the 6th international conference on computational linguistics and intelligent text processing (CICLing), Mexico City, pp 486–497
141. Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. In: Proceedings of the conference on human language technology and empirical methods in natural language processing (HLT), Vancouver, pp 347–354
142. Witten IH, Frank E, Hall MA (2011) Data mining: practical machine learning tools and techniques, 3rd edn. Morgan Kaufmann, Burlington
143. Xia R, Zong C (2010) Exploring the use of word relation features for sentiment classification. In: Proceedings of the 23rd international conference on computational linguistics (COLING), pp 1336–1344
144. Xia R, Zong C, Li S (2011) Ensemble of feature sets and classification algorithms for sentiment classification. J Inf Sci 181(6):1138–1152
145. Ye Q, Zhang Z, Law R (2009) Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. Expert Syst Appl 36(3):6527–6535
146. Yu L, Liu H (2004) Efficient feature selection via analysis of relevance and redundancy. J Mach Learn Res 5(1):1205–1224
147. Zhang C, Zeng D, Li J, Wang FY, Zuo W (2009) Sentiment analysis of Chinese documents: from sentence to document level. J Am Soc Inf Sci Technol 60(12):2474–2487
148. Zhang L, Liu B (2011) Identifying noun product features that imply opinions. In: Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies, vol 2, no 1, pp 575–580
149. Zhu J, Xu C, Wang HS (2010) Sentiment classification using the theory of ANNs. J China Univ Posts Telecommun 37(1):58–62

Index

B. Agarwal, N. Mittal, *Prominent Feature Extraction for Sentiment Analysis*, Socio-Affective Computing 2, DOI 10.1007/978-3-319-25343-5