Aspect-based sentiment analysis for social recommender systems.

CHEN, Y.Y.

2019

Aspect-based Sentiment Analysis for Social Recommender Systems

Chen, Yoke Yie

A thesis submitted in partial fulfilment
of the requirements of the Robert Gordon University
for the degree of Doctor of Philosophy

May 2019

Abstract

Social recommender systems harness knowledge from social content, experiences and interactions to provide recommendations to users. The retrieval and ranking of products, using similarity knowledge, is central to the recommendation architecture. To enhance recommendation performance, having an effective representation of products is essential. Social content such as product reviews contain experiential knowledge in the form of user opinions centred on product aspects. Making sense of these for recommender systems require the capability to reason with text. However, Natural Language Processing (NLP) toolkits trained on formal text documents encounter challenges when analysing product reviews due to their informal nature. This calls for novel methods and algorithms to capitalise on textual content in product reviews together with other knowledge resources. In this thesis, methods to utilise user purchase preference knowledge inferred from the viewed and purchased product behaviour are proposed to overcome the challenges encountered in analysing textual content.

This thesis introduces three major methods to improve the performance of social recommender systems. First, an effective aspect extraction method that combines strengths of both dependency relations and frequent noun analysis is proposed. Thereafter, this thesis presents how extracted aspects can be used to structure opinionated content enabling sentiment knowledge to enrich product representations. Second, a novel method to integrate aspect-level sentiment analysis and implicit knowledge extracted from users' product purchase preferences analysis is presented. The role of sentiment distribution and threshold analysis on the proposed integration method is also explored. Third, this thesis explores the utility of feature selection techniques to rank and select relevant aspects for product representation. For this purpose, this thesis presents how established dimensionality reduction approaches from text classification can be employed to select a subset of aspects for recommendation purposes. Finally, a comprehensive evaluation of all the proposed methods in this thesis is presented using a computational measure of '*better*' and Mean Average Precision (MAP) with seven real-world datasets.

Declaration of Authorship

I declare that I am the sole author of this thesis and that all verbatim extracts contained in the thesis have been identified as such and all sources of information have been specifically acknowledged in the bibliography. Parts of the work presented in this thesis have appeared in the following publications:

- Chen, Y. Y., Wiratunga, N. and Lothian, R. (2017). Effective Dependency Rule-based Aspect Extraction for Social Recommender Systems. In *Pacific Asia Conference on Information Systems.* Association for Information Systems (AIS). (**Chapter 4**)

- Chen, Y. Y., Ferrer, X., Wiratunga, N., and Plaza, E. (2014). Sentiment and preference guided social recommendation. In *Case-Based Reasoning Research and Development*, pages 79–94. Springer. (**Chapter 5**)

- Ferrer, X., Chen, Y. Y., Wiratunga, N., and Plaza, E. (2014). Preference and sentiment guided social recommendations with temporal dynamics. In *Research and Development in Intelligent Systems XXXI*, pages 101–116. Springer. (**Chapter 5**)

- Chen, Y. Y., Ferrer, X., Wiratunga, N., and Plaza, E. (2015). Aspect selection for social recommender systems. In *Case-Based Reasoning Research and Development*, pages 60–72. Springer. (**Chapter 6**)

- Chen, Y. Y., Wiratunga, N. and Lothian, R. (2018). Integrating Selection-based Aspect Sentiment and Preference Knowledge for Social Recommender Systems. *Online Information Review: Social Recommender Systems.* Reviewers decision: Accepted for publication. (**Chapter 6**)

Acknowledgements

I would like to express my deepest appreciation to my supervisors Professor Nirmalie Wiratunga and Dr. Rob Lothian for their unwavering support, advice, and encouragement throughout my time as a research student at RGU. You have always offered me the best support and advice whenever I needed them.

I would like to thank RGU for offering training, infrastructure to conduct my research and also giving me the opportunities to attend conferences and summer school to present my research and network with other researchers in the same field. This had helped me immensely in my development as a researcher and also my research work.

I would also like to thank my colleagues in N426, staff members from the SIS research group as well as SICSA research communities for providing insightful feedback on my research. I also appreciate the help and support from the administrative team in CSDM and also the IT support team, who played a role in making this research possible. I am thankful to Jeremie Clos for his unwavering support throughout my PhD journey.

Finally, I would like to express my gratitude to my family for their support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recommender systems are computer systems that provide suggestions for products that are likely to be liked by a particular user. In order to identify the preferred products for a user, recommender systems predict or compare the utility of products before providing a ranked list of recommended items to users. A simple and non-personalised recommendation algorithm that recommends only the most popular products was first proposed to provide this ranking. The popularity of a product can be measured using the product sales or its ratings. The assumption for this popularity-based approach is that a popular product, which is liked by many users, will also most-likely be preferred by other users (Ricci et al., 2015). However, this popularity approach will present to any user a predefined, fixed list of products regardless of the user's preferences, which can lead to disappointment in the recommender system (Cremonesi et al., 2010).

Providing personalised recommendations requires the recommender systems to have information about the products and the users' preferences. Traditionally, users' preferences can be inferred in two ways: collaborative filtering and content-based approaches. Collaborative filtering (CF) employs user ratings to infer user preferences (Koren et al., 2009, Sarwar et al., 2001, Su and Khoshgoftaar, 2009), where ratings of an existing user community with similar preferences to the target user drive recommendation. However, CF models are particularly sensitive to the data sparsity and the cold-start problems (Esparza et al., 2011).

The collaborative filtering approach is built on the knowledge of product and user preferences, and does not put user preferences in context when providing recommendations. Context-aware recommender systems leverage contextual information in addition to the knowledge of users and products to improve recommendation performance. For example, the recommender system not only know how much a given user liked a specific product (e.g. ratings), but also the contextual information in which the product was purchased by the user (e.g. a temporal context would be "Saturday afternoon", while a physical context would be "bookings made from a smartphone"). Similar to CF, context-aware approaches also suffer from sparsity problems (Haruna et al., 2017, Yujie and Licai, 2010). To overcome the limitations of collaborative filtering and context-aware approaches, content-based approaches build product profiles using product descriptions in order to model user preferences from their past purchase preferences (Lops et al., 2011, Pazzani and Billsus, 2007).

The dawn of the social web created many new opportunities for content-based recommendation algorithms to improve recommendation performance thanks to the vast amounts of publicly available social information. Recommender systems that leverage social information to improve recommendation performance are called social recommender systems (Guy, 2015). Social recommender systems that use social information to build product representation are a form of content-based recommendation. Typically, the standard approach in content-based recommender systems is to use a set of relevant keywords that appear in the product description to build a product representation. However, these approaches fail to consider users' purchase experiences and preferences which are key to their purchase decisions. To overcome these weaknesses, social information such as social tags, microblogs, explicit social relationships (social network) and users' click behaviour are now being used in social recommenders to improve recommendation accuracy. However, this social knowledge is less useful when data sparsity is high as a result of users only tagging or commenting on a limited number of items or having fewer interactions with each others.

Product reviews are a form of consumer feedback where consumers express opinions about aspects of a product. In the context of product reviews and product recommendation, the term aspect denotes both components and the characteristics of a product[1]. Consider the following review example:

'The camera has a good lens but the battery is not very durable.'

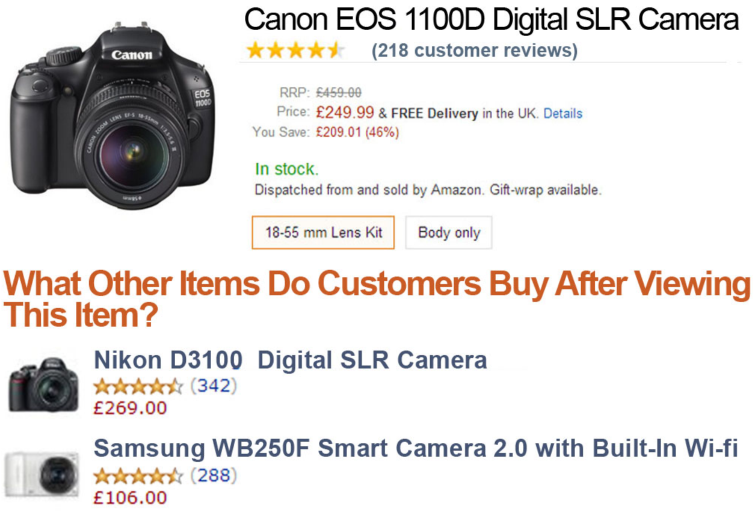
Here, the reviewer expresses conflicting opinions about two product aspects of a camera - aspect *lens* connotes positive sentiment whilst aspect *battery* is negative. Such fine-grained opinions are important, in that they explain a consumer's preferences that drive their purchase decisions and should naturally influence the workings of recommender systems. However, in practice, not all users can be expected to rate products after a purchase and highly priced products tend to receive limited reviews from a single user (Jindal and Liu, 2008, Xie et al., 2012). To overcome this limitation, additional sources of social knowledge such as users' purchase preferences are required to further improve recommendation performance.

Consider a typical product recommendation scenario on an e-commerce website in Figure 1.1. Here, there are product information (e.g. an image of a camera product, name of the product), explicit user preferences such as average user's ratings, product price and also information generated or derived from user interactions (e.g. reviews and items that users actually buy after viewing this camera). Specifically, it can be observed that there is implicit knowledge in the form of user preferences. Here, preferences refer to purchase preferences of users over viewed products. Whilst purchase preferences approximate users' preferences they do not explain why a product is being purchased. Therefore, strategies to quantify users' interest in specific aspects of a given product are needed.

There has been a steady increase in social recommender systems research which is evidenced by the growth of research papers in this area in the last decade (see Table 1.1). These figures were obtained from SciVal[2], a research trend analysis tool developed by

[1] In the literature, aspects are also referred to as features (Liu, 2015)

[2] https://www.elsevier.com/en-gb/solutions/scival



Canon EOS 1100D Digital SLR Camera

(218 customer reviews)

RRP: ~~£459.00~~
Price: £249.99 & FREE Delivery in the UK. Details
You Save: £209.01 (46%)

In stock.
Dispatched from and sold by Amazon. Gift-wrap available.

What Other Items Do Customers Buy After Viewing This Item?

Nikon D3100 Digital SLR Camera
(342)
£269.00

Samsung WB250F Smart Camera 2.0 with Built-In Wi-fi
(288)
£106.00

FIGURE 1.1: Product information.

Elsevier[3], using keywords such as "recommender systems", "users's review" and "implicit feedback". Despite the existence of research work in exploiting product reviews and implicit feedback to recommend products, social recommender systems remain an open research field. Linguistic nuances that are caused by the informal nature of social media text make it challenging to automatically extract product aspects from reviews and assess their sentiment value. Furthermore, the available forms of implicit feedback are abundant but not knowledge-rich. Therefore, strategies to utilise both knowledge sources to improve recommendation performance are needed.

Year	**Users' Reviews**	**Implicit Feedback**
2008	21	7
2009	37	18
2010	44	10
2011	60	21
2012	68	31
2013	85	27
2014	109	51
2015	119	42
2016	173	56
2017	211	67
2018	242	78

TABLE 1.1: Number of Research Papers Related to Recommender Systems

[3] A world-leading provider of scientific, technical and medical information products and services

1.1 Related Research Fields

Social recommender systems research has over the years built upon techniques from different research fields (see Figure 1.2). Extracting social knowledge from social media requires techniques from Aspect-based Sentiment Analysis (ABSA), Natural Language Processing (NLP) and Feature Selection (FS). Encoding this social knowledge in a meaningful format that a computer system can parse and make use of in order to perform recommendation tasks involves the research fields of Textual Case-based Reasoning (TCBR) and Information Retrieval (IR). The rest of this section highlights how the advances of each of the aforementioned research fields have influenced the research of social recommender systems.

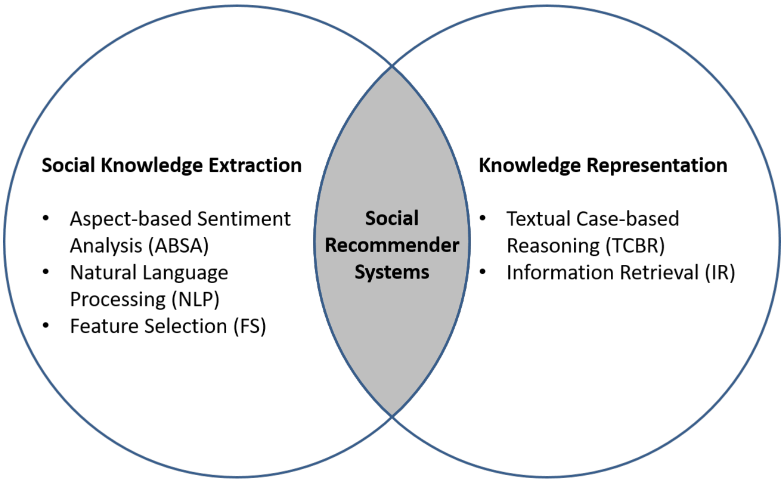

Social Knowledge Extraction
• Aspect-based Sentiment Analysis (ABSA)
• Natural Language Processing (NLP)
• Feature Selection (FS)
Social Recommender Systems
Knowledge Representation
• Textual Case-based Reasoning (TCBR)
• Information Retrieval (IR)

Figure 1.2: Related Research Fields

Textual Case-based Reasoning is a sub-field of case-based reasoning, which focuses on solving new problems by using the solutions of other similar problems using knowledge sources in a textual format (Weber et al., 2005). Item representation and similarity-based algorithms from TCBR approach have made a significant contribution to recommender systems research. A recommender system that adopts a textual case-based approach is a form of content-based recommendation that defines each product using a set of product aspects such as *price* and *color* (Bridge et al., 2005, Lorenzi et al., 2005, Smyth, 2007) and retrieves products based on similarity between query and candidate product. The most common approach in feature-based representation is called a vector space model where each product is represented as a vector in n-dimensional vector space (Christopher et al., 2008). In order to compare the similarity of two products, the similarity of the vector space is measured using similarity metrics.

There are a number of distance and similarity metrics implemented in recommender systems such as Euclidean (Chen and Wang, 2013), Jaccard (Zhang and Pennacchiotti, 2013) and Cosine (Dong et al., 2014, Zhang and Pennacchiotti, 2013). Cosine similarity is established as the most popular technique in measuring similarity due to its performance in producing the most accurate results (Jannach et al., 2010). There are a few reasons why cosine similarity performs better than other metrics. Firstly, cosine similarity measures the similarity of vectors with respect to the origin. This metric is a measurement of orientation and not magnitude. Secondly, Euclidean distance measures the distance between two points in the vector space. This means that two vectors that have the same orientation (contain the same aspects) can have a high Euclidean distance between them because the common terms have huge differences in weights (or magnitude). Thirdly, Jaccard is not a vector-based distance measure. It measures the similarity between products based on the number of shared aspects between them. The main drawback of this approach is that the relative importance of the aspects is not represented. The work in this thesis is inspired by the case-based approach and proposes a preference-based aspect weighting approach for product representation. To generate recommendations, products are retrieved based on the similarity of the query and candidate product.

Aspect-based Sentiment Analysis is a task that determines the orientation of sentiment expressed on each aspect in a sentence (Liu, 2012). Generally, social recommender systems that analyse user's opinion in product reviews for recommendation employ aspect-based sentiment analysis techniques. There are two main tasks in aspect-based sentiment analysis: aspect extraction and sentiment classification. Aspect extraction can be seen as an information extraction task that extracts aspects that the reviewer refers to in a given review. There are two main approaches for aspect extraction: supervised and unsupervised. A supervised approach is least favoured due to the challenges in obtaining ground truth data to evaluate the performance of new algorithms. Existing work has shown that unsupervised dependency relation-based approaches outperform both frequent noun and supervised approaches (Poria et al., 2014, Qiu et al., 2011). This is partly due to important infrequent aspects that get typically filtered-out by these approaches but will be extracted by the dependency relation rules. Since dependency-based methods extract aspects by means of syntactic relations between pairs of words in a sentence, they are not restricted to frequent aspects only. Representing products based on product aspects and user sentiments rely on techniques from aspect based sentiment analysis (ABSA) research. Exploiting dependency relations between words in a sentence is a well established method in ABSA research. This thesis explores the most efficient dependency-based approaches for aspect extraction.

Sentiment analysis at the aspect level takes into account the distance of the sentiment word and the target aspect in a sentence (Liu, 2015). Once the target aspect has been identified, a sentiment aggregation function is applied to determine the overall sentiment value of an aspect for a given product. SmartSA is a lexicon-based sentiment classification system for social media text. Evaluation results show that SmartSA performed significantly better than a state-of-the-art sentiment classification system for social media text, SentiStrength, with a reported average F-Score of 70.4 (Muhammad et al., 2016). SmartSA is relevant to the work presented in this thesis because the recommendation strategies proposed in this thesis capitalise on social media text to provide recommendation. Since the main contribution in this thesis is not in the area of sentiment analysis, SmartSA is applied in the works in this thesis to determine the sentiment score of sentiment-bearing words.

Natural Language Processing is an area of research that explores the automated processing of human language to perform predefined tasks. Therefore, this research area is highly relevant to social recommender systems that capitalise on textual features in online reviews to recommend products. Specifically, NLP techniques such as tokenization, lemmatization, part-of-speech (POS) tagging and dependency relation annotation are essential techniques required in the aspect extraction task. These techniques are typically found in popular NLP toolkits such as Stanford CoreNLP[4] and GATE[5]. However, standard NLP approaches face difficulties when applied to social media text due to its informal nature. For instance, Stanford CoreNLP parser cannot recognise negation with the omission of apostrophe in the sentence such as *"I dont like the screen of the camera"*. The key to extracting meaningful aspects from social text is through the analysis of the syntactic structure of text. One of the major methods in representing the syntactic structure of natural languages is dependency relations (Potisuk, 2010). Therefore, in this thesis, the limitations of the NLP toolkits were taken into account and the focus is shifted toward a strategy to select dependency relations for aspect extraction.

Information Retrieval is a research field that is concerned with identifying and retrieving a set of documents that are relevant to a given user information need. Text representation and link analysis are IR techniques which have been widely applied in recommender systems. Basic IR-inspired text representation such as the vector space model with a Term Frequency and Inverse Document Frequency (TF-IDF) weighting scheme is a standard item representation approach in content-based recommender systems. When using this approach in a review-based social recommender system, each product is viewed as a document that made up of a set of aspects contained in the reviews (Esparza et al., 2011). The aspects are weighted based on how informative they are with respect to the product. However, this approach is agnostic of user's opinion on each aspect and thus has a disadvantage in recommending products that have a better quality (in terms of user's sentiment), which is the area that this thesis will explore.

In search engine, link analysis is the analysis of hyperlinks and graph structures for web search. In the context of recommender systems, this approach has been applied to recommend scholarly articles to users by analysing citation patterns. Such approach is also important for product recommendation, for instance, by comparing sentiment values of aspects between users' viewed and purchased products. This thesis proposes a preference graph that is generated from a set of viewed-purchased product pairs in order to elicit user's preferences.

[4] http://nlp.stanford.edu/software/dependencies_manual.pdf

[5] https://gate.ac.uk

Feature Selection is the process of selecting a subset of relevant features to build models that solve machine learning problems such as classification and clustering (Cai et al., 2018). In a text classification problem, the subset of features is expected to be sufficient in correctly predicting the class of an unseen text document. Despite the substantial research work done on feature selection for machine learning algorithms, the application of feature selection techniques in recommender systems is under-explored (Ronen et al., 2013). In the context of social recommender systems, feature selection techniques can be applied to evaluate the relevance of product aspects in reviews. The feature selection techniques can be categorised into supervised and unsupervised approaches. Supervised feature selection requires labelled training data. In contrast, unsupervised approaches select features without any labelled data. A major limitation in the applicability of supervised learning is that user-generated content (e.g. product reviews) lacks labelled training data and it is costly to obtain human labelled data. This area is explored in this thesis in the form of a comparison of the recommendation performance between supervised and unsupervised feature selection techniques.

1.2 Research Motivation

Unlike CF, content-based approaches are able to explicitly list content features to explain why an item was recommended. As mentioned earlier in this chapter, the standard content-based approach is to use a set of relevant keywords that appear in the product description to identify similar products to recommend. However, these approaches fail to consider users' purchase experiences and preferences which are key to their purchase decisions. To overcome these weaknesses, users' purchased experiences written in product reviews are used to enhance recommender performance. However, relying on user-generated reviews for product representation has limitations:

- The effectiveness of using a dependency-based approach in aspect extraction is evident in existing work (Poria et al., 2014, Qiu et al., 2011). There are 47 dependency relations defined in the Universal Dependencies[6] for English and every sentence can trigger more than one dependency relation. However, previous work selects a subset of the dependency relation rules without providing information on how the rules were chosen (Moghaddam and Ester, 2012, Poria et al., 2014, Qiu et al., 2011). It is important to have this information in order to select relevant dependency rules as the irrelevant rules can result in erroneous aspects being extracted.

- Social media text is characterised by a diverse vocabulary. A product may have hundreds of aspects which are not equally important to consumers when making a purchase decision (Zha et al., 2014). This becomes a challenge when recommending products to new users (cold-start users) when their preferences are not known by the system. Therefore, additional sources of social knowledge are needed to help estimate the importance to different aspects. Because of the abundance of social knowledge related to those purchase decisions, methods to combine different sources of social knowledge to infer aspect importance are sought.

- Natural language processing (NLP) based product aspect extraction techniques that rely on Part-Of-Speech (POS) tagging and syntactic parsing are known to be less robust when applied to informal text (Owoputi et al., 2013). As a result, a large number of spurious content can be incorrectly extracted as aspects. However, most previous work ignores the selection of aspects and thus limits the potential of using reviews for recommendation.

In order to address the aforementioned limitations in relation to integrating social knowledge in recommender systems, this thesis explores the following research questions:

- Which dependency relations are most relevant to extract aspects that improve recommendation performance?

- Which explicit and implicit knowledge sources can be integrated and what impact do they have on recommendation performance?

[6] http://universaldependencies.org/en/dep/index.html

- Can feature selection methods used for dimensionality reduction in classification be used to select relevant aspects for social recommendation?

1.3 Research Objectives

This thesis investigates and defines new methods for social recommender systems. The overall aim is to improve recommendation performance by extracting relevant aspects and combining explicit and implicit social knowledge. To achieve this aim, the following objectives are defined:

1. Develop a dependency-based product aspect extraction technique that improves recommendation performance.
2. Develop a product ranking algorithm using social knowledge captured from product reviews and users purchase preferences.
3. Investigate the utility of feature selection techniques to select relevant aspects for product representation.
4. Conduct a comprehensive evaluation of all developed strategies.
5. Create a dataset consisting of product details from multiple product categories (Cameras, Laptops, Tablets, Phones, Printers, Mp3 players and TV) and the corresponding users' purchase preferences.

Objective 1 is achieved using explicit social knowledge and objective 2 and 3 are achieved by combining both explicit and implicit social knowledge.

1.4 Contributions

An overview of the social recommendation process and the main contributions of this thesis is shown in Figure 1.3. The final outcome of the recommendation process is a list of recommended products that are ranked on the basis of a $ProductScore$ with respect to a given query product. Central to this ranking is the computational model of aspect-level user preferences derived from product reviews with dominant products inferred from the preference graph. To generate a product representation for each product, aspects are extracted from reviews using an aspect extraction approach. Given the extracted aspects, aspect describing sentiment words are identified from the reviews in order to compute aspect sentiment scores using a sentiment classification system. A weighted aspect-level sentiment analysis is proposed and the weights of aspects are learned by comparing the sentiment difference between node pairs in the preference graph. In order to explore the utility of feature selection techniques in improving recommendation performance, an alternative approach to recommendation is proposed. In this approach, instead of using all the extracted aspects in generating product representation, the extracted aspects go through the aspect selection process where relevant aspects are retained to generate a product representation.

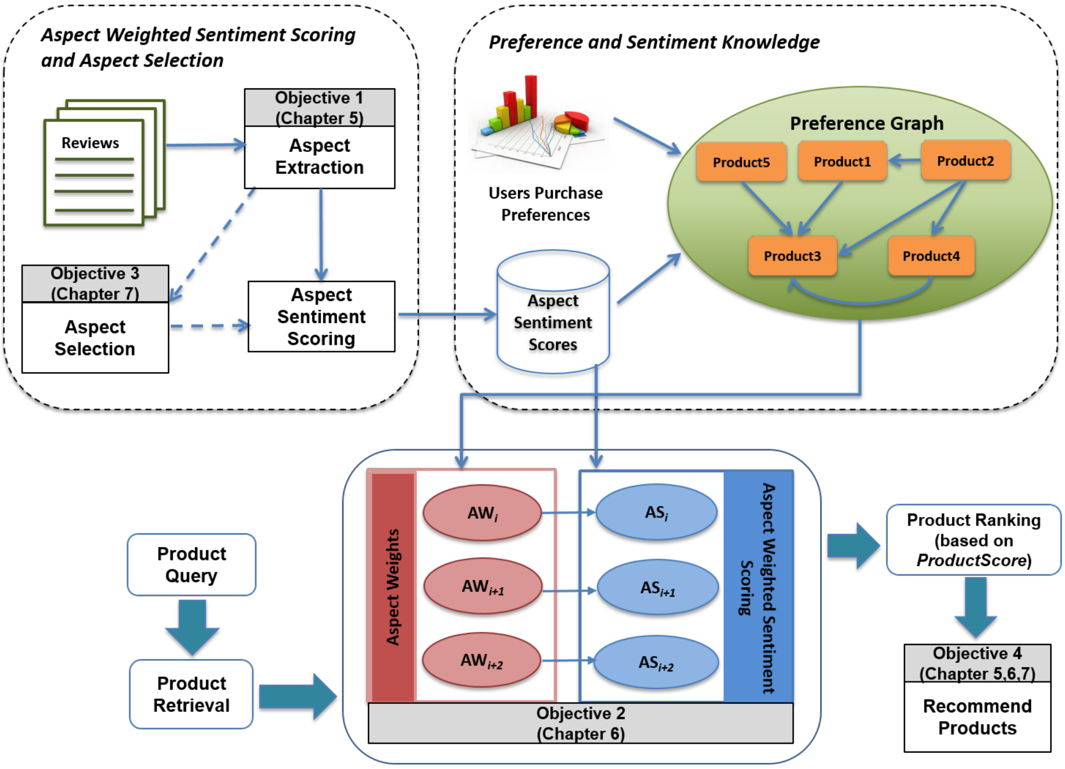


FIGURE 1.3: The Social Recommendation Process that Illustrates the Steps where the Research Objectives and Contributions of the Thesis Lies

The main contributions of this thesis are the following:

- The first main contribution of this research is the development of an informed aspect extraction approach that combines the strengths of both dependency relations and rule-based frequent noun approaches. The selection of dependency rules is performed based on their ability to frequently relate noun and sentiment terms. The proposed approach is compared to state-of-the-art dependency-based approaches in a recommendation setting. It is important to describe the process of dependency rules selection in order to avoid selecting irrelevant rules that can result in the extraction of erroneous aspects and a detrimental effect on recommendation performance. Evaluation results show that recommendation performance of the proposed informed selection of dependency relations approach improves when combined with the rule-based frequent noun approach. An analysis of the aspects extracted by each aspect extraction approach suggests that when applying the dependency relations approach to extract aspects, frequency pruning is required to remove spurious aspects. This further emphasises the importance of combining the rule-based frequent noun approach with dependency relations approach when performing aspect extraction.

- The second main contribution is the development of an aspect weighting approach that integrates social knowledge from product reviews and users' purchase preferences. Specifically, the proposed approach combines sentiment knowledge from product reviews and preference knowledge from users' purchase preferences. The insight is that aspects that are likely to have influenced the users' purchase decisions can be identified through the preference relationships modelled in the preference graph. Evaluation results show that combining users' product purchase preferences and sentiment knowledge can effectively improve recommendation performance. Specifically, setting a sentiment threshold when computing aspect preference difference score gives the best performance overall. In order to consider the distribution of sentiments of an aspect in the recommendation algorithm, Gini and Wilson score were applied. Results on applying the Gini score in the recommendation algorithm show no improvement. An analysis of the Gini scores in the dataset shows that there is little social agreement on the sentiment expressed on majority of the product aspects. Thus, Gini has little effect on recommendation performance. In contrast, results on applying Wilson score are mixed where performance improvement is observed only on specific datasets. Further analysis on the Wilson scores in the dataset shows that the limited occurrence of unique aspects limits the opportunity of the Wilson score to improve recommendation performance.

- The third main contribution is the development of a recommendation method that integrates feature selection techniques to select important aspects for product representation. Specifically, in a supervised feature selection technique, this work proposes to use user ratings as proxy class labels. This approach is useful considering labelled training data is not always available for user-generated content and it is expensive to create. A comparative study of four feature selection technique suggests that the unsupervised feature selection approach, document frequency (DFREQ), gives the best performance in majority of the datasets. The performance of supervised approaches such as information gain (IG) and Chi-squared was poor due to the class imbalance problem. However, in the absence of the class imbalance problem, the results demonstrated that supervised approaches performs better than unsupervised approaches. Analysing the difference in aspect subset size shows that users used different terminology to refer to the same aspect. Therefore, in order to achieve a lower number of aspects, the semantic similarity between aspects needs to be considered. Further experiments were conducted to assess the effect of applying the aspect weights on the selected aspects. Results show that the aspect weighting benefits unsupervised feature selection approaches the most. When using supervised approaches, recommendation performance is better without integrating the aspect weights.

- The fourth main contribution of this research is creating real-world datasets from seven different product categories. These datasets include product information, product reviews, user ratings, best seller rank and users purchase preferences. Specifically, the users purchase preferences contain a list of products that other consumers buy after viewing the product. In this thesis, the seven datasets are used to develop the proposed recommendation strategies and evaluate the recommender systems.

The proposed recommendation approaches in this thesis does not require individual user preferences to provide recommendations to users. Therefore, the proposed sentiment and preference-guided strategy for product recommendation is a feasible solution to recommend products to new users (e.g. cold-start users) even if their preferences are not known by the recommender system. Further, a key research implication from the proposed recommender system is its ability to provide explanations on the recommended products to users, due to its reliance on aspect sentiment to recommend products. Being able to justify a recommendation using aspects, weights, and user opinions provides a first step towards future work in providing users with explanations for their recommended products.

1.5 Thesis Overview

This chapter provides an overview of social recommender systems, which exploit social knowledge from product reviews and users preference knowledge. The research fields that are related to this research and the motivation of the research have been discussed. The research objectives as well as the main contributions of this research have been given. The rest of the thesis is outlined below.

Chapter 2 and 3 present a review of the literature related to the work presented in this thesis. Chapter 2 discusses recent works on aspect-based sentiment analysis with a particular focus on dependency rules and frequent noun approaches. Further, feature selection techniques and existing approaches in aspect selection are discussed. Thereafter, in Chapter 3, an overview of the state-of-the-art approaches to social recommendation is given and the different sources of social knowledge applied in social recommender systems are discussed. Evaluation methodology, evaluation metrics and datasets applied in recommender system research are also discussed.

Chapter 4 presents the background of this research. This includes the baseline algorithms and the details of the evaluation datasets, evaluation methodology and evaluation metrics employed are introduced.

Chapter 5 presents a comparative study of different aspect extraction approaches for recommendation tasks. This includes frequent noun approaches and the dependency-based models. The best practices when extracting aspects from dependency relations generated by Stanford CoreNLP are also explored. The proposed dependency selection process and the heuristic rules applied are discussed. Thereafter, the aspect sentiment scoring algorithm that is used to score a product for ranking is discussed. The chapter ends with a comparative study on how different aspect extraction approach affects recommendation performance.

Chapter 6 presents the proposed aspect weighting algorithm and aspect weighted sentiment scoring algorithm. This chapter is divided into four sections. The first section discusses the available explicit and implicit social knowledge and how implicit knowledge is modelled into a preference graph. The second section presents the integration of sentiment knowledge and users' preferences to formalise the proposed aspect weighting algorithm and aspect weighted sentiment scoring algorithm. The following section discusses the insights from the evaluation datasets to illustrate that the aspects extracted from Chapter 5 are adequate for product comparison. Finally, the last section presents the comparative study of the proposed approach together with the baseline approaches.

Chapter 7 improves the representation developed in Chapter 5 and 6 by selecting a subset of relevant aspects for product representation. This chapter starts by discussing the motivation for aspect selection in product representation. Thereafter, the supervised and unsupervised feature selection techniques that are applied in this research are also discussed. Specifically, the discussion involves how product ratings are utilised to define class labels for supervised feature selection approaches when human annotated class labels are not available. Finally, this chapter presents a comparative study of different feature selection techniques.

This thesis concludes in Chapter 8 with a summary of the main contributions. The limitations of the proposed approaches explored in this thesis are identified and future directions of this research to overcome them are discussed.

Chapter 2

Aspect-based Sentiment Analysis

Social recommender systems that analyse product reviews for recommendation generally employ methods from aspect-based sentiment analysis (ABSA). The idea is to use extracted aspects to represent products whereby the strength of sentiment in either positive or negative direction forms the aspect values. As such information extraction methods are particularly relevant here as aspects must be extracted from textual content. To understand the aspect extraction approach used in this thesis, a review of the state of the art in aspect extraction and of relevant methods from sentiment classification is presented. Differentiating useful aspects from a large set of extracted aspects is likely to impact recommendation judgements. Following the success of feature selection strategies in related areas such as text classification, this chapter also explores selection heuristics that are likely to be transferable to aspect selection.

2.1 Aspect-based Sentiment Analysis

Opinions can be expressed on any product, service or person. The target of an opinion is referred to as an entity. An entity can have a set of aspects. For example, *iPhone* is an entity that has a set of aspects such as *battery* and *screen*. In the field of sentiment analysis, aspects are often referred to as features, product features or opinion targets (Liu, 2015). An opinion is a positive or negative view about an entity or an aspect of an entity expressed by an opinion holder. The positivity, negativity and neutrality characterises opinion orientation (or sentiment polarity in sentiment analysis literature) whereby no opinion is considered neutral sentiment. Formally, an opinion is a quintuple (Liu, 2012):(e_i, a_{ij}, oo_{ijkl}, h_k, t_l); where e_i is the name of the entity i, a_{ij} is the j^{th} aspect of entity e_i, oo_{ijkl} is the k^{th} opinion orientation of the opinion expressed on aspect a_{ij} of entity e_i, by the opinion holder h_k at time t_l. An aspect can be explicitly mentioned in a review or implied through other expressions (implicitly). For example, *screen* in the sentence "*The screen is wide*" is an explicit aspect. In contrast, implicit aspect expressions often identified through adjectives (Su et al., 2008). For example, *expensive* in the sentence "*The IPhone 6 is really expensive*" is an implicit aspect that refers to the aspect *price*. Research in the area of aspect extraction has mainly focused on extracting explicit aspects. One of the main reasons is that there is no standard dataset available to test and evaluate new implicit aspect extraction algorithms (Tubishat et al., 2018). Ths focus of this thesis is to extract explicit aspects.

A positive opinion in a review does not necessarily mean that the author is positive about everything; similarly, with a negative opinion. Instead, it is a summarised indicator of the general orientation tendency. In a typical product review scenario, mining user sentiments at the review level and sentence level is useful as it provides more granular analysis of orientation. However, such information is insufficient to support purchase decisions without also knowing the target aspects that the opinion is expressed on. Therefore, aspect extraction is crucial to drive sentiment analysis for social recommender systems as it provides insight into purchase decisions.

There are two main tasks in aspect-based sentiment analysis: aspect extraction and sentiment classification. Here, aspect extraction techniques are organised into three different categories: frequent noun approach, dependency relations model and supervised learning. The sentiment classification task is usually performed after aspect extraction; as such, aspect extraction techniques will be discussed first, and sentiment classification second.

Prior research indicates that product aspects are generally nouns or noun phrases (Nakagawa and Mori, 2002). Most approaches discussed will start off with extracting aspects based on this criterion. However, only extracting nouns will lead to many spurious aspects. Therefore, the selection of relevant aspects will also be explored.

2.1.1 Frequent Noun Approach

The frequent noun approach identifies product aspects that are expressed by noun and noun phrases from a large corpus of product reviews. Pioneering work on aspect extraction from product reviews using the frequent noun approach was presented by Hu and Liu (2004). They apply the Apriori algorithm to identify a list of product aspects. There are two main steps in the Apriori algorithm. First, it finds all frequent itemsets from a set of transactions that satisfy a user-specified minimum support. In the second step, it generates rules from the discovered frequent itemsets. For aspect extraction task, only the first step is used, that is to find frequent itemsets, which are the candidate aspects. In addition, only frequent itemsets with three words or fewer are kept as it is assumed that product aspects contain no more than three words. Here, a single sentence forms the transaction and items consists of nouns or noun phrases identified by a Part-of-Speech (POS) tagger. Therefore, an itemset is a set of nouns or noun phrases that occur together in a sentence. An itemset is defined as frequent if it appears above a specified support threshold. Further, to identify genuine aspects from the list, two pruning methods are applied to remove candidate aspects that do not frequently appear together and those that are redundant. To evaluate the approach, a dataset that consists of 5 electronic products reviews crawled from Amazon.com and Cnet.com (2 digital cameras, 1 cellular phone, 1 mp3 player and 1 dvd player) are used and manually labelled with aspects (if there are any). The proposed method achieves an average precision and recall of 0.72 and 0.80 respectively. The reason for retaining high frequency noun and noun phrases is that when reviewers comment on different aspects of a product, the vocabulary they use is limited. Therefore, aspects that are frequently mentioned are deemed more important and so assumed to be more genuine.

A major shortcoming of association mining is that it generates many aspects that are not genuine. This is especially true in product reviews where authors will describe their experience or an event in reviews without providing any opinion. Furthermore, some of the nouns that are extracted as aspects are subject to parsing errors. While Hu and Liu (2004) perform heuristic pruning to eliminate non-aspect terms, others have focused on improving the pruning method. Popescu and Etzioni (2007) first extract a nouns and noun phrases list from product reviews and thereafter prunes the list with a frequency threshold. The remaining candidate aspects in the list are evaluated using the Pairwise Mutual Information (PMI) between the candidate aspect and associated extractor pattern. For instance, typical patterns for the camera class are, "a of camera", "a comes with camera", "a is part of camera" where a is the product aspect. The purpose of having these extraction patterns is to find components of cameras on the Web. Given a product aspect a and pattern d. The PMI score is computed as follows:

$$PMI(a,d) = \frac{hits(a \wedge d)}{hits(a)hits(d)} \tag{2.1}$$

where $hits(a)$ and $hits(d)$ is Web search engine hit counts for aspect a and pattern d respectively. Similarly, $hits(a \wedge d)$ is the hit counts for the co-occurrences of a and d. When testing this approach using the same dataset[1] used in Hu and Liu (2004), it was observed that combining PMI with the frequent noun approach gave an average precision score of 0.88 which is about a 22% improvement, with just a 3% reduction in recall.

Although the PMI approach provides significant improvement, the domain specific extraction patterns suggest that this approach is not easily transferable to other domains (e.g. travel, hotels, restaurants). To overcome this problem, Moghaddam and Ester (2010), Li et al. (2009), Htay and Lynn (2013) and Rana and Cheah (2017) uses POS patterns to identify product aspects. For example, a common pattern such as noun adjective allows the identification and extraction of the associated noun. A major shortcoming of the frequent noun approach is that technical aspects of a camera such as *aperture* (opening of a camera through which light travels) may only appear in reviews written by professional photographers and as such is likely to be missed out by being infrequent. Accordingly, methods that are less reliant on frequency are needed to address such complexities.

[1] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

2.1.2 Dependency Relations Model

An opinion in product reviews consists of two key components: a target (aspect) and a sentiment on the target (Liu, 2012). Therefore, there is a relationship between an aspect and the sentiment expressed about the aspect. Relation based approaches exploit this relationship to extract aspects and their associated sentiments. The intuition is that since sentiment words are easy to find, relationships can be used to identify new aspects. Hu and Liu (2004) use a manually crafted sentiment lexicon[1] and the "nearest" function approximates dependency relations between noun or noun phrases (aspect) and the sentiment words that describe aspects. The sentiment lexicon has about 10,000 sentiment words that consist of positive and negative sentiments. For a working example of this approach, consider the following sentences:

"The picture is amazing."

From the sentiment lexicon, *amazing* appears to be a sentiment word because it will typically have a high sentiment polarity score, then *picture* (a noun) is extracted as an aspect. This method is straightforward and easy to implement. However, the manually crafted sentiment lexicon is not comprehensive, and instead a public lexicon such as SentiWordNet (Esuli and Sebastiani, 2006) with over 200,000 sentiment word sense pairs is more useful. Thus Hu and Liu (2004) approach limits the opportunity to identify new aspects.

An alternative to adjacent based sentence analysis that does not rely on sentiment lexicon to find product aspects, is to use a dependency parser to determine the semantic relationships between words. This is more likely to generate opinion phrases accurately than methods that consider the proximity of words alone (Moghaddam and Ester, 2012). A dependency parser provides a list of dependency relations that describe the relationship of words in a sentence. For a given sentence, grammatical relations (also called syntactical relationships and typed dependencies[2]) is a set of triples, $Rel\{w_1, w_2\}$, each of which is composed of a dependency relation Rel and the words w_i and w_j from the sentence forming the dependency relation. In product reviews, opinions expressed by

[2] In Stanford CoreNLP, a dependency relation is referred as typed dependencies representation.

users are often centred around target aspects. The key idea here is that an opinion has a target and there are often explicit syntactical relationships between an opinion word and its target aspect. By exploiting this relation, the words in a dependency relation can be used to identify the product aspects and the sentiment words that describe the aspects. Figure 2.1 shows an example of the dependency relations output from Stanford CoreNLP[3] using the previous example.

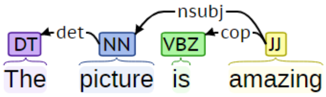


FIGURE 2.1: Example of Dependency Relations Representation from Stanford CoreNLP

Here, the sentence is tagged with its POS and the relation between words are linked with a dependency relation. For example, the noun (NN) *picture* depends on the adjective (JJ) *amazing* through ***nsubj***. As of 2017, there are 47 dependency relations in the Stanford CoreNLP[4]. Previous work has either used these relations individually or in combination to extract aspects and sentiment words. Therefore, two categories of relations that encompass all possible relations between two words in sentences can be used: direct dependency and indirect dependency (Qiu et al., 2011). A direct dependency indicates that one word depends on the other word without any additional words in the dependency path. Table 2.1 summarises a list of frequently used direct dependency relations together with the extracted aspect and sentiment word (Bancken et al., 2014, Zhuang et al., 2006). In each example, the word in bold is the extracted aspect word and the underlined word is the sentiment word.

An indirect dependency indicates that one word depends on another word through an additional word. Consider the first example given in Table 2.2. Here, the adjective *nice* is dependent on the noun *case* through verb *looks*. Therefore, *case* is extracted as the aspect and *nice* is the sentiment word that describes the aspect *case*. The list of just three direct dependencies in Table 2.1 is insufficient to extract all potential aspects and sentiments. Therefore, Qiu et al. (2011) proposed a propagation method to find all possible aspects and sentiments. They start off with a small set of seed sentiment words

[3] http://stanfordnlp.github.io/CoreNLP/
[4] http://universaldependencies.org/en/dep/index.html

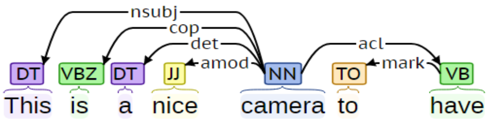
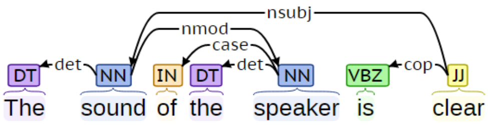
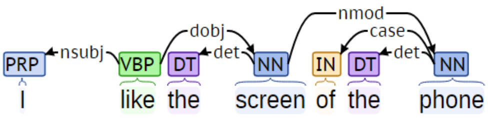
Dependency Relations	**Examples**	**Output**
Adjectival modifier (amod). An adjectival modifier of an NP (Noun Phrase) is any adjectival phrase that serves to modify the meaning of the NP	 nsubj cop det amod acl mark DT VBZ DT JJ NN TO VB This is a nice camera to have	*amod*(**camera**, nice)
Nominal subject (nsubj). A nominal subject relation is a noun phrase which is the syntactic subject of a clause.	 nsubj nmod case det det cop DT NN IN DT NN VBZ JJ The sound of the speaker is clear	*nsubj*(clear, **sound**)
Direct object (dobj). The direct object of a VP (Verb Phrase) is the NP which is the (accusative) object of the verb.	 nmod dobj case nsubj det det PRP VBP DT NN IN DT NN I like the screen of the phone	*dobj*(like, **screen**)

TABLE 2.1: Examples of Direct Dependency.

Dependency Relations	**Examples**	**Output**
$nsubj(w_1, w_2) + dobj(w_1, w_3)$	The camera case looks nice	*nsubj*(looks, **case**) *dobj*(looks, nice)
$amod(w_1, w_2) + conj\_and(w_1, w_3)$	The camera has great zoom and resolution	*amod*(zoom, great) *conj_and*(**zoom**, **resolution**)
$nsubj(w_1, w_2) + advmod(w_1, w_3)$	The battery works well	*nsubj*(works, **battery**) *advmod*(works, well)

TABLE 2.2: Examples of Indirect Dependency

for extraction from Hu and Liu (2004)'s sentiment lexicon. Then, for each sentiment word they implement the following rules to extract aspects and its sentiments:

1. Use a predefined set of dependency relations (e.g. *amod*, *nsubj*) to extract aspects using seed sentiment words. For example, given *amod*(picture, nice) and that *nice* is a seed sentiment word, *picture* is extracted as an aspect if its POS tag is a noun.

2. Extract aspects using extracted aspects in step 1. This step involves dependency relations such as ***conj*** (conjunction) and ***compound*** (compound nouns). For

example, given that *picture* is an aspect and *picture* is found in *compound* (quality, picture). Then, *quality* is extracted as an aspect if it is a noun.

3. Extract sentiment words using extracted aspects in step 2. For example, step 2 determines *quality* to be an aspect and it depends on the adjective *good* through ***amod***. Then, *good* is extracted as a sentiment word.

4. Extract sentiment words using both given and extracted sentiment words. Similar to step 3, this step involves ***conj*** and ***compound***. For example, in *conj*(easy, good), given that *good* is a sentiment word, *easy* is also extracted as sentiment word.

5. Repeat step 2 to 4 until there are no new aspects or sentiment words to be found.

The key idea of this approach is that with each known sentiment word, more aspects can be found, and vice versa. During the search process, sentiment words are considered to be adjectives and aspects are nouns or noun phrases. The resulting list of aspects is pruned as follows:

- Pruning based on clauses. When the aspects occur in the same clause and are not connected by a conjunction (e.g. and, or), one of the aspects that occur less frequently will be removed.

- Pruning of other product names and store names where the consumer purchased the product.

- Identifying target phrases and global pruning. Target phrases are identified by combining each extracted aspect word with Q consecutive nouns right before and after the aspect word, and K adjectives before the target aspect. Here Q and K is set as 2 and 1 respectively. Thereafter, target phrase that appear only once is removed.

The approach outperforms the initial method proposed by Hu and Liu (2004) and Popescu and Etzioni (2007). Several approaches have been introduced to improve this baseline with minimal performance improvement reported in Liu et al. (2015), Xu et al.

(2013), Zhang et al. (2010b) and Kang and Zhou (2017). It is not surprising that significant improvement is hard to achieve using bootstrapping methods. First, this method could extract many nouns/noun phrases that are not aspects and therefore does not scale well to large datasets. This is because during propagation, adjectives that are not opinionated will be extracted as opinion words. Increasingly, more and more noise is introduced to the expanding lexicon and aspect sets.

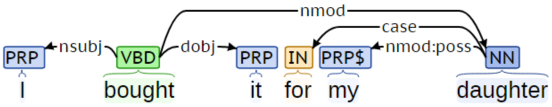


Figure 2.2: Dependency Relations – Example 2

Furthermore, using dependency relations can still lead to spurious term extraction. For instance, in Figure 2.2, noun (NN), *daughter*, is related by the dependency relation ***nmod*** (Nominal Modifier) with the verb (VB) *bought*. Notice that although *daughter* is a noun, it is not a valid aspect. This example demonstrates how the application of shallow heuristics can lead to erroneous extractions of aspects, which will invariably have a detrimental effect on recommendation performance. To overcome this limitation, previous work selects a subset of the dependency relation rules without providing information on how the rules were chosen (Bancken et al., 2014, Moghaddam and Ester, 2012, Poria et al., 2014). However, it is important to have the information in order to select relevant dependency rules as the irrelevant rules can result in erroneous aspects. This is because to get a good coverage of aspects, many dependency rules need to be used (Schouten and Frasincar, 2016). Therefore, selection strategies are needed to identify the relevant set of rules. It can be observed from the example given in Figure 2.2 that the use of sentiment knowledge would have shown that *daughter* is not a valid aspect as it is not related to a sentiment-bearing verb. Similarly, frequency information may also have conveyed that *daughter* is an infrequent noun in camera reviews. Based on this observation, it is important to include sentiment knowledge in dependency relation selection as well as frequency information to remove infrequent nouns.

Sentiment lexicons such as SentiWordNet and Hu and Liu's sentiment lexicon are commonly used to identify aspects. Poria et al. (2014) proposed a rule-based aspect extraction algorithm called SenticNet aspect parser[5] using SenticNet as the sentiment lexicon to extract aspects. SenticNet is built by integrating multiple common knowledge and common sense knowledge bases (e.g. DBPedia (Bizer et al., 2009), ConceptNet (Speer and Havasi, 2012), Cyc (Lenat and Guha, 1989) and Open Mind Common Sense) to produce a large semantic graph (Cambria et al., 2014). Each node in the graph represents a concept (a word that can be found in free text). To determine whether a concept is an emotion related word, the polarity score of the concept is computed using the emotion categorisation model proposed by Cambria et al. (2012). Concepts which are highly linked to emotion nodes (emotion words) are retained. As a result, there are 30,000 emotion related concepts available in SenticNet. Therefore, it is expected that non sentiment-bearing words that exist in SenticNet are linked to emotion words.

The SenticNet aspect parser capitalises on common-sense knowledge and a set of manually defined dependency relation rules to identify potential aspects from review sentences (Poria et al., 2014). Evaluation results have shown that the SenticNet aspect parser outperforms the frequency noun approach proposed by Hu and Liu (2004) and Popescu and Etzioni (2007) as well as the dependency propagation approach by Qiu et al. (2011) as described previously in page 23 and 24. Accordingly, Figure 2.3 shows the flowchart of the rule-based algorithm applied in the SenticNet aspect parser (Poria et al., 2014). The list of rules applied in the aspect parser extract explicit and implicit aspects. However, as described in Section 2.1, the focus of this work is extracting explicit aspects. Therefore, rules that extract explicit aspects are considered relevant to the work presented in this thesis. Sample output of the rule-based algorithm is shown in Table 2.3.

SenticNet aspect parser applies Stanford CoreNLP parser to generate the list of dependency representation for each sentence. As shown in Figure 2.3, the extraction of aspects is triggered when a term t_y is in a *nsubj* relationship with a term t_x (e.g. $nsubj(t_x, t_y)$). Given the list of dependency representations generated by the Stanford CoreNLP parser

[5] http://sentic.net/demos/#aspect

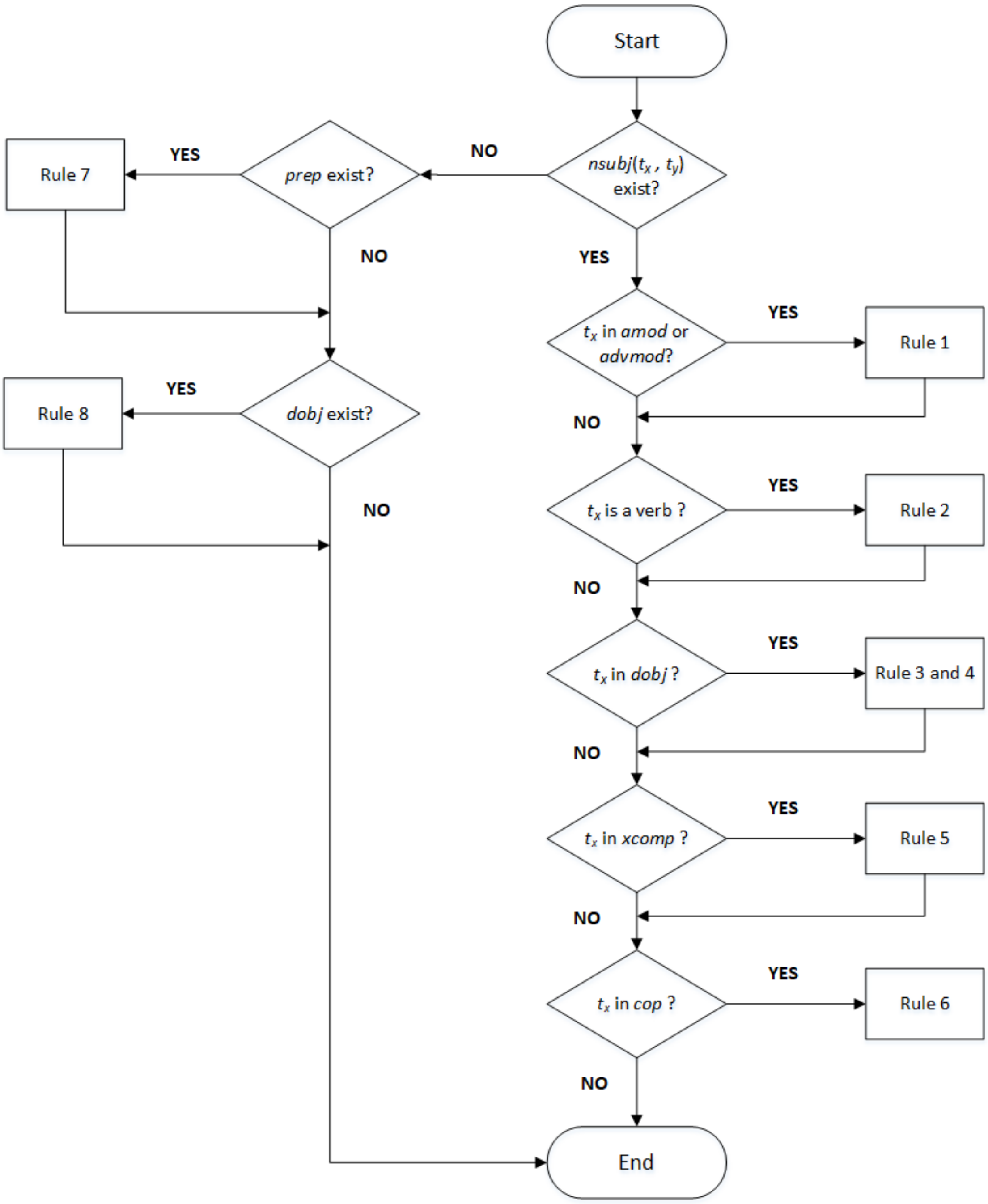


FIGURE 2.3: Main Flowchart

for each example, the first example sentence in Table 2.3 triggers Rule 1 in Figure 2.4. The list of dependency representations for the first example contains *nsubj* and *amod*. Given that *camera* is a noun, *it* is in a subject noun relation with *camera*. Here, *camera* is connected to *nice* in the relation *amod* and *nice* is found in SenticNet. Hence, *camera* is extracted as an aspect.

The SenticNet aspect parser uses a combination of dependency relations with manually

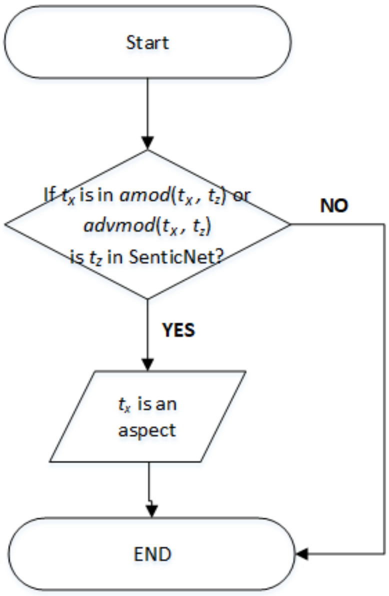

Start
If t_x is in *amod*(t_x, t_z) or *advmod*(t_x, t_z) is t_z in SenticNet?
NO
YES
t_x is an aspect
END

FIGURE 2.4: Rule 1

Examples	**Dependency Representations (dR)**	**Rules**	**Aspects**

It is a nice camera	nsubj(camera, it)
	amod(camera, nice)
		Rule 1	**camera**

I like the size of the screen	nsubj(like, I)
	dobj(like, size)
	prep_of(size, screen)
		Rule 3 and 4	size
			size, screen

Not to miss the battery of the camera	dobj(miss, battery)	Rule 8	**battery**

TABLE 2.3: Example use of SenticNet Aspect Parser

defined rules to extract aspects. Unlike the approach proposed by Moghaddam and Ester (2012) which extracts aspects using a predefined set of dependency relations, the SenticNet aspect parser adopts extraction rules similar to Qiu et al. (2011) by first extracting aspects using a set of predefined dependency relations and iteratively expanding such aspect set using heuristic rules. For instance, consider the second example in Table 2.3 which triggers Rule 3 and 4 in Figure 2.5. Here *like* is in a subject noun relationship with *I* and *like* is also in a *dobj* relation with *size*. Given that *size* is a noun that exists in SenticNet, *size* is extracted as an aspect. Next, *size* is connected to *screen* in a *prep* relation and *screen* is a noun. Therefore, *screen* is also extracted as an aspect. Finally, sentences with no subject noun relationship trigger Rule 7 or 8. In the third example, *nsubj* is not available and *miss* is in the *dobj* relation with *battery*. This triggers Rule

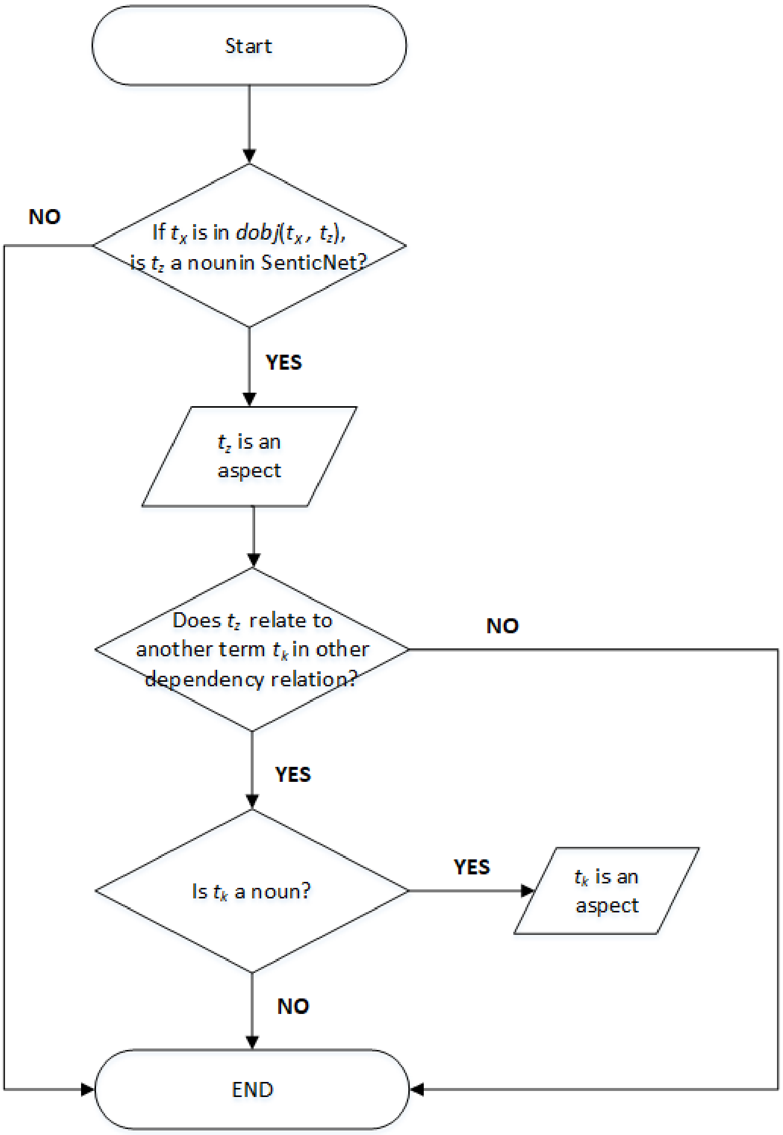

Start
If t_x is in *dobj*(t_x, t_z), is t_z a nounin SenticNet?
NO
YES
t_z is an aspect
Does t_z relate to another term t_k in other dependency relation?
NO
YES
Is t_k a noun?
YES
t_k is an aspect
NO
END

Figure 2.5: Rule 3 and 4

8 in Figure 2.6 and *battery* is extracted as an aspect. A full set of extraction rules 1 to 10 is shown in Appendix B.

Once all the aspects were extracted using Rule 1 to 8, Rule 9 and 10 are triggered to extract multi-word aspects (e.g. picture quality) and aspects which were not extracted in Rule 1 to 8 using the list of extracted aspects. First, Rule 9 is triggered to extract additional aspects as shown in Figure 2.7. For every aspect in the extracted aspect list, if an aspect term a is in a *cc* or *conj* relation with another term t_m, then t_m is an aspect. Thereafter, Rule 10 is triggered to extract multi-word aspects (Figure 2.7). If there is a *compound*(t_m, t_n) relation and t_n is an aspect, then t_n - t_m is a multi-word aspect and t_n is removed from the extracted aspect list. One limitation observed in Rule 10 is that removing the single term aspect may risk loosing important aspects. For instance, consider '*picture*' is an important aspect in camera products but it will be removed when multi-word aspects such as '*picture quality*' are identified. This will cause applications such as recommender systems to lose important knowledge on a product which may lead to poor recommendation performance.

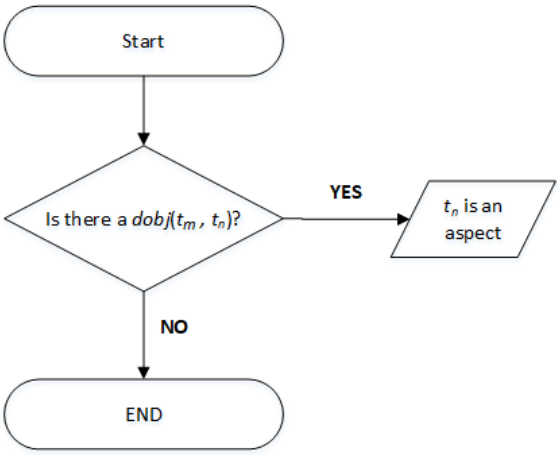


FIGURE 2.6: Rule 8

2.1.3 Supervised Machine Learning

The task of identifying aspects and sentiment words can be stated as a sequential learning problem. Thus, classical sequential learning methods such as Hidden Markov Model and Conditional Random Fields are commonly used for this task.

2.1.3.1 Hidden Markov Model (HMM)

A hidden Markov model represents probability distributions over a sequence of observations (Rabiner, 1989). This model have been applied in POS tagging and named-entity recognition (NER) problems (Liu, 2015). In aspect extraction, the words and phrases in review text can be seen as observations and aspects or sentiment words are labels and act as the underlying states. To extract product aspects and sentiment words, Jin et al. (2009) proposed a lexicalised HMM approach. They integrate linguistic features such as POS and surrounding contextual clues of words into automatic learning. To build the

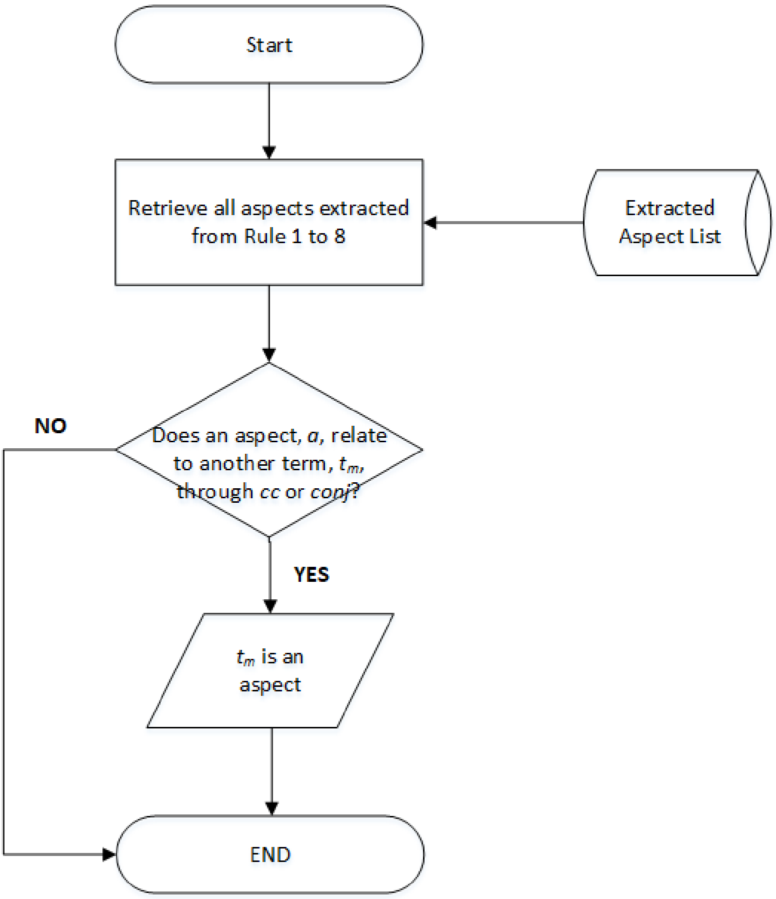

Start
Retrieve all aspects extracted from Rule 1 to 8
Extracted Aspect List
Does an aspect, *a*, relate to another term, t_m, through *cc* or *conj*?
NO
YES
t_m is an aspect
END

Figure 2.7: Rule 9

learning model, they defined two categories of tag sets to tag each sentence representing the patterns between aspect and opinion words. An observable state is represented by a pair $(word_i, POS(word_i))$ where $POS(word_i)$ represents the POS of $word_i$. Therefore, the task was to find the appropriate sequence of tags that maximize the conditional probability. An observed disadvantage of hidden Markov models is that their linear sequence structure is not adequate for review text where an aspect can appear 2 to 3 words before or after its associated sentiment word in a sentence. An undirected sequence model is needed to address this limitation.

2.1.3.2 Conditional Random Fields (CRF)

CRF are a discriminative undirected graph model that focus on the conditional distribution $p(y|x)$ over a hidden sequence y given a sequence of observations x (Lafferty et al.,

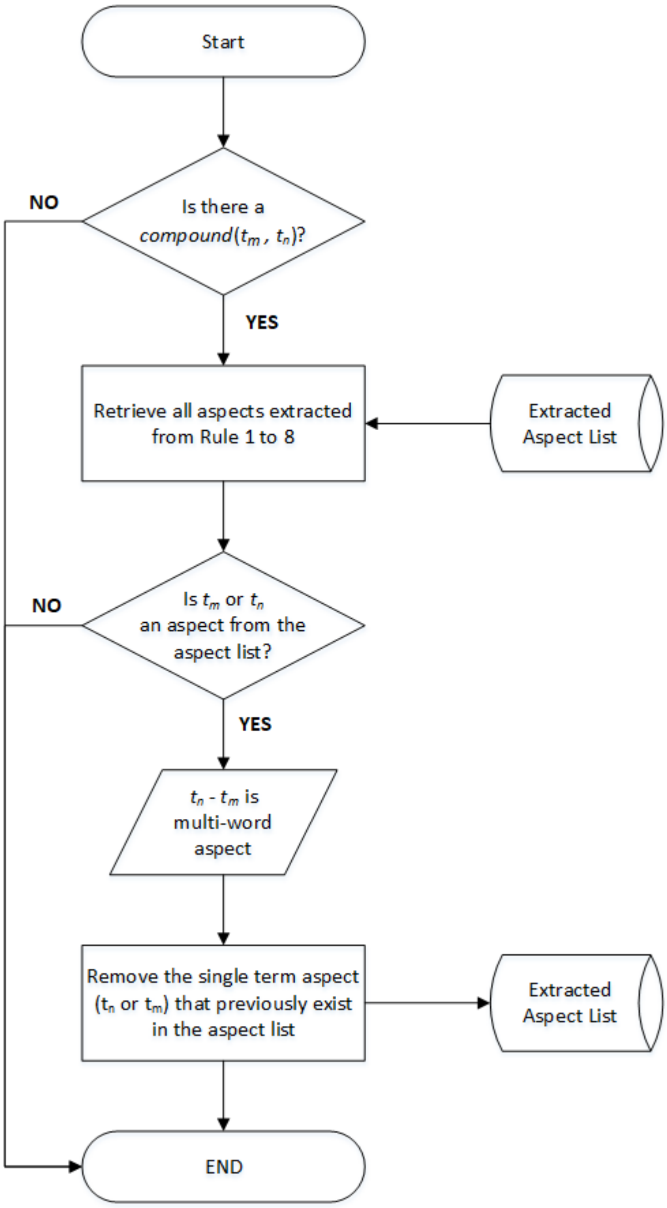

Start
Is there a compound(t_m , t_n)?
NO
YES
Retrieve all aspects extracted from Rule 1 to 8
Extracted Aspect List
Is t_m or t_n an aspect from the aspect list?
NO
YES
t_n - t_m is multi-word aspect
Remove the single term aspect (t_n or t_m) that previously exist in the aspect list
Extracted Aspect List
END

Figure 2.8: Rule 10

2001). Therefore, unlike HMM which are tied to a linear-sequence structure, CRF can be arbitrarily structured.

Jakob and Gurevych (2010) utilised CRF to extract aspects from reviews. They employed multiple features to form the feature function for their CRF approach:

- Token - The string of current token.
- POS - The part-of-speech of current token.

- Short dependency path - Label tokens which have direct dependency relation to opinion expression.
- Word distance - Label the closest tokens for each opinion expression in a sentence.
- Opinion Sentence - Label tokens which occur in a opinion bearing sentence.

The possible class labels are represented following the IOB scheme: *I-Target*, identifying the continuation of a target, O for other (non-target) tokens, and *B-Target* identifying the beginning of an opinion target. Each review sentence is modelled as a linear chain CRF that is based on an undirected graph. Here, each node in the graph corresponds to each word in the sentence and edges connected to adjacent tokens as they appear in the sentence.

CRF are known for being unable to capture long-range dependencies (i.e. there are many words occurring between the aspects and its sentiment word) and it has been shown in Qiu et al. (2011) that many aspects and sentiment word pairs have long-range dependencies. To overcome this limitation, Li et al. (2010) proposed a structure-aware CRF model. They integrate two variations of CRFs namely Skip-tree CRF and Tree-CRF to extract aspects and sentiments by providing a list of aspects as input seeds. The design of their model takes into account the long distance dependency with conjunctions (e.g. and, but) and deep syntactic dependencies of aspects. Unlike the classical CRF model which only depends on word sequence in learning, their proposed approach exploits the linguistic structure of aspects.

Recent work using deep neural networks show performance improvement on aspect extraction (Poria et al., 2016, Wang et al., 2017). However, supervised approaches require annotated training data from social media text which is often difficult to acquire. Therefore, recommendation algorithms targeting real-world datasets favour unsupervised approaches over supervised approaches. Besides supervised machine learning methods, topic modelling has been a popular approach in identifying implicit topics and sentiment words (Liu, 2015). However, the aim of these are usually not to extract a list of aspects from reviews, but instead to categorise aspects. Therefore, this approach tends to identify coarse-grained topics or aspects that correspond to the discussed entity, but does not necessarily make sense as products aspects.

2.1.4 Aspect Sentiment Classification

Sentiment classification assigns a positive or negative label to opinionated documents, paragraphs or sentences. There have been extensive studies in sentiment classification with both supervised and unsupervised lexicon-based approaches. Unlike classical sentiment classification, aspect sentiment classification aims to consider the aspect in a sentence during classification. Therefore, the approaches proposed for aspect sentiment classification tend to differ from traditional sentiment classification in order to take into account of the sentiment of the relevant aspects only.

2.1.4.1 Supervised Learning Approach

Sentiment classification can be seen as a text classification problem. Therefore, existing supervised machine learning approaches can be used to predict the sentiment class of unlabelled documents. However, algorithms such as Support Vector Machine (SVM) and Naive Bayes classification proposed by Pang et al. (2002) for sentiment classification are no longer sufficient for aspect sentiment classification. This is because features used in training the algorithm are agnostic of aspects and thus unable to determine which aspect is the product aspect of the target of the sentiment expressed.

To address this limitation, Yu et al. (2011) utilises short reviews in the 'Pros' and 'Cons' section of the full review to train an SVM classifier. Since short reviews are labelled with polarity by the author, they use sentiment terms in these reviews as features and represent each short review as a feature vector. These sentiment terms are found using a sentiment lexicon provided by the MPQA project[6]. To link the sentiment term to its aspect, sentiment words positioned within a distance of five from the aspect in a parse tree are selected.

[6] https://mpqa.cs.pitt.edu/lexicons/

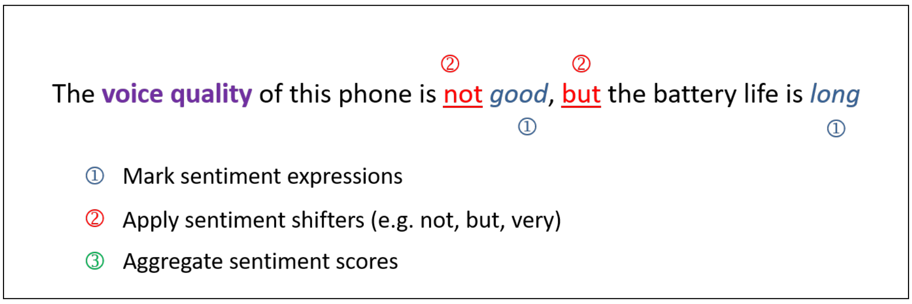
Although the aspect sentiment association problem is addressed by the aforementioned techniques, labelled data is hard to acquire and time consuming. In addition, a sentiment classifier that is trained in one domain often performs poorly in another. Although transfer learning can be a good alternative, accuracy achieved on the new domain tends to be lower (Muhammad et al., 2016).

2.1.4.2 Lexicon-based Approach

Lexicon-based approaches avoid some of the issues observed with supervised learning. Here, the polarity of a sentiment-bearing word is ascertained by looking up a sentiment lexicon (a list of words associated to a set of scores for each sentiment orientation). In recent years, SentiWordNet (Esuli and Sebastiani, 2006) has become the primary source for aspect-based sentiment analysis due to its high coverage of English terms and fine-grained sentiment information.

The typical steps of aspect sentiment classification shown in Figure 2.9 assume that aspects are first extracted, such as '*voice quality*'. In step 1, sentiment words are identified using a sentiment lexicon. Here, words '*good*' and '*long*' are marked as sentiment words and the polarity for '*good*' is determined as positive but not for '*long*' since it's a context dependent sentiment word. Next, any sentiment shifters that appear around the sentiment expressions are marked. Some of the examples of common sentiment shifters are *not, very* and *but* (Muhammad et al., 2016). Once shifters are identified, Step 2 will turn the '*good*' sentiment expression to negative owing to the negation word '*not*'. Then, words that indicate conflicting information need to be handled as they often change sentiment orientation. A sentence containing the conflicting word (e.g. *but*, *however*) will switch the sentiment orientation before and after the conflicting word (Liu, 2015). Therefore, the sentiment after the '*but*' clause is found to be positive. In the final step, an aggregation function is applied to determine the final sentiment score for an aspect.

One major concern in aspect-based sentiment classification is how to link the aspect and its sentiment word. This is because a sentence may consist one or more aspects and sentiments. Therefore, to determine the sentiment word that modifies the aspect, similar



The **voice quality** of this phone is not ② *good* ①, but ② the battery life is *long* ①

① Mark sentiment expressions

② Apply sentiment shifters (e.g. not, but, very)

③ Aggregate sentiment scores

Figure 2.9: Aspect Sentiment Classification Steps

word adjacency approach used by Yu et al. (2011) was proposed. Hu and Liu (2004) and Moghaddam and Ester (2010) consider the nearest adjective word to an aspect in the same sentence as the sentiment word. Zhu et al. (2009) proposed a multi-aspect sentence segmentation model where each sentence is segmented with each segment consisting of an aspect. Then the polarity of each segment is determined using a sentiment lexicon. Although this approach works well, it has been observed that the sentiment word that describes an aspect may appear far from the aspect.

An alternative is to exploit syntactic dependencies of sentiment words and their aspects which has the advantage of capturing long range dependencies (as discussed in Section 2.1.2). There are two methods to ascertain the sentiment scores. First, sentiment word can be identified from the dependency relations generated by the Stanford CoreNLP parser[7]. Aspects are typically nouns, whilst adjectives, adverbs and verbs tend to capture sentiment (Hu and Liu, 2004, Popescu and Etzioni, 2007). Therefore, sentiment-rich adjectives, adverbs and verbs that relate to nouns are extracted from dependency relations. To illustrate the first approach, consider the examples in Figure 2.10 and 2.11 together with word dependencies.

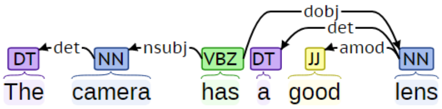


Figure 2.10: Dependency Relations - Example 3

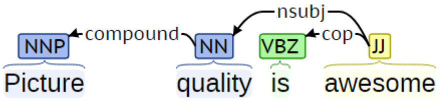


Figure 2.11: Dependency Relations - Example 4

[7] http://stanfordnlp.github.io/CoreNLP/

In Figure 2.10, the dependency relation, ***amod***, correctly connects noun (NN), *lens*, with adjective (JJ), *good*, to extract the target aspect, *lens*. However, not all dependency relations relate a noun with sentiment word. For instance, in Figure 2.11, noun (NN), *picture*, is related by the dependency relation, ***compound***, with the noun (NN) *quality*, but notice that ***compound*** relates the nouns but is not rich in sentiment. Therefore, the second approach is more suited this work, in that it finds the nearest sentiment-rich word to an aspect in the same sentence as the target sentiment word. Such word can appear on the left or right side of an aspect. Therefore, the heuristic that the sentiment word with the minimum distance (minimum number of words) from the aspect as the target sentiment word is preferable.

The key advantage of a lexicon-based approach is its domain independence. It does not require hand labelled text to train a model as required by a supervised learning approach. Although it does require initial effort in building the knowledge base, once this is built, it can be easily extended by updating or inserting new knowledge.

2.1.5 Application of Aspect Extraction in Recommender Systems

The aim of applying an aspect extraction algorithm in recommender systems is to build a product representation using product aspects given in reviews. Most previous works implement this task by manually gathering a list of key product aspects from external sources such as consumer reports, e-commerce websites or previous research work (Ganu et al., 2013, Jamroonsilp and Prompoon, 2013, Yates et al., 2008, Zhang et al., 2010a). However, product reviews are characterised by a diverse vocabulary where reviewers refer to the same product aspect in different ways (e.g. picture quality versus photo quality). To overcome this limitation, an additional step is required to examine whether the extracted word from the review is similar to the manually identified key aspects.

A viable alternative to the manual approach is to apply the frequent noun approach discussed in Section 2.1.1 (Dong et al., 2016, Liu et al., 2013, Muhammad et al., 2015). Instead of identifying product aspects using association rule mining, Dong et al. (2016) proposed to extract aspects using a combination of shallow NLP and statistical methods, primarily by combining ideas from Justeson and Katz (1995) and the frequent noun approach. In this approach, there are two types of aspects:

- aspects that appear as bigrams.
- aspects that appear as a single noun.

Bigrams are extracted as aspects by detecting the following part-of-speech patterns:

- an adjective followed by a noun where the adjective is not found in a sentiment lexicon[8]. This condition is put in place to avoid extracting single noun aspects that are preceded by an adjective word. For instance, *excellent lens* refers to the single noun aspect, *lens*, and not a bigram aspect.
- a noun followed by another noun (e.g. *battery life*).

Single nouns extracted from reviews are often not related to product aspects. For example, single nouns like *day* and *family* are not aspects. One solution proposed by Hu and Liu (2004) is to remove single nouns that are rarely associated to sentiment words. The intuition is that nouns that frequently co-occur with opinion words are likely to be genuine aspects. Therefore, nouns that have a frequency greater than a fixed threshold are retained. This aspect extraction approach is applied in a recommendation algorithm that recommends helpful reviews.

Aspect extraction based on dependency relation rules extracts aspects by means of syntactic relations between pairs of words in a sentence. Chen and Wang (2013) applied dependency relations in extracting aspects and sentiment words to consider both the overall ratings and aspect-level sentiment values as extracted from product reviews to identify reviewers' preference using a Latent Class Regression model. However, the list of dependency relations that they used in their work were not provided. Moghaddam and Ester (2012) proposed 9 dependency rules (rule du_1 to du_9) to extract aspects as shown in Figure 2.12. Here N is a noun, A an adjective, V a verb and $\langle h, m \rangle$ is a candidate phrase. Figure 2.13 shows the flowchart that applies rule du_1 to du_4 to extract candidate

[8] A collection of positive and negative sentiment words which frequently appear in social media text (Hu and Liu, 2004)

phrase. Once all the candidate phrase were extracted using rule du_1 to du_4, rule du_5 to du_9 in Figure 2.14 are used to extract additional aspects based on the extracted candidate phrase $\langle h, m\rangle$.

$$\begin{aligned}
&du_1 : amod(N, A) \rightarrow \langle N, A\rangle,\\
&du_2 : acomp(V, A) + nsubj(V, N) \rightarrow \langle N, A\rangle,\\
&du_3 : cop(A, V) + nsubj(A, N) \rightarrow \langle N, A\rangle,\\
&du_4 : dobj(V, N) + nsubj(V, N') \rightarrow \langle N, V\rangle,\\
&du_5 : \langle h_1, m_1\rangle + compound(h_1, N) \rightarrow \langle N + h_1, m_1\rangle,\\
&du_6 : \langle h_1, m_1\rangle + compound(N, h_1) \rightarrow \langle h_1 + N, m_1\rangle,\\
&du_7 : \langle h_1, m_1\rangle + conj(h_1, h_2) \rightarrow \langle h_2, m_1\rangle,\\
&du_8 : \langle h_1, m_1\rangle + conj(m_1, m_2) \rightarrow \langle h_1, m_2\rangle,\\
&du_9 : \langle h_1, m_1\rangle + neg(m_1, not) \rightarrow \langle h_1, not + m_1\rangle,
\end{aligned}$$

Figure 2.12: Dependency Relation Rules.

Examples of how these rules apply to product review sentences are shown in Table 2.4.

Examples	**Dependency Representations (dR)**	**Rules (du_i)**	**Aspects**

The camera lens is good	cop(good, is)
	nsubj(good, lens)
	compound(lens, camera)
		du_3
		du_5
			(lens, good)
			(**camera lens**, good)

This camera has good screen and lens	amod(screen, good)
	conj(screen, lens)
		du_1
		du_7
			(**screen**, good)
			(**lens**, good)

The picture quality is awesome	nsubj(awesome, quality)
	cop(awesome, is)
	compound(quality, picture)
		du_3
		du_5
			(quality, awesome)
			(**picture quality**, awesome)

Table 2.4: Example use of Dependency Relation Rules

The Stanford CoreNLP parser generates dependency representation (dR) for each review sentence. Given the list of dR and the list of dependency rules for the first example sentence in Table 2.4, rule du_3 is triggered (*good* is an adjective and *is* is a verb) as shown in Figure 2.13:

$$du_3\text{: } cop(good, is) + nsubj(good, lens) \rightarrow \langle lens, good\rangle.$$

Here. the extracted candidate phrase is $\langle lens, good\rangle$. Next, rule du_5 to du_9 in Figure 2.14 is applied to extract additional candidate phrase. It can be observed that a Noun

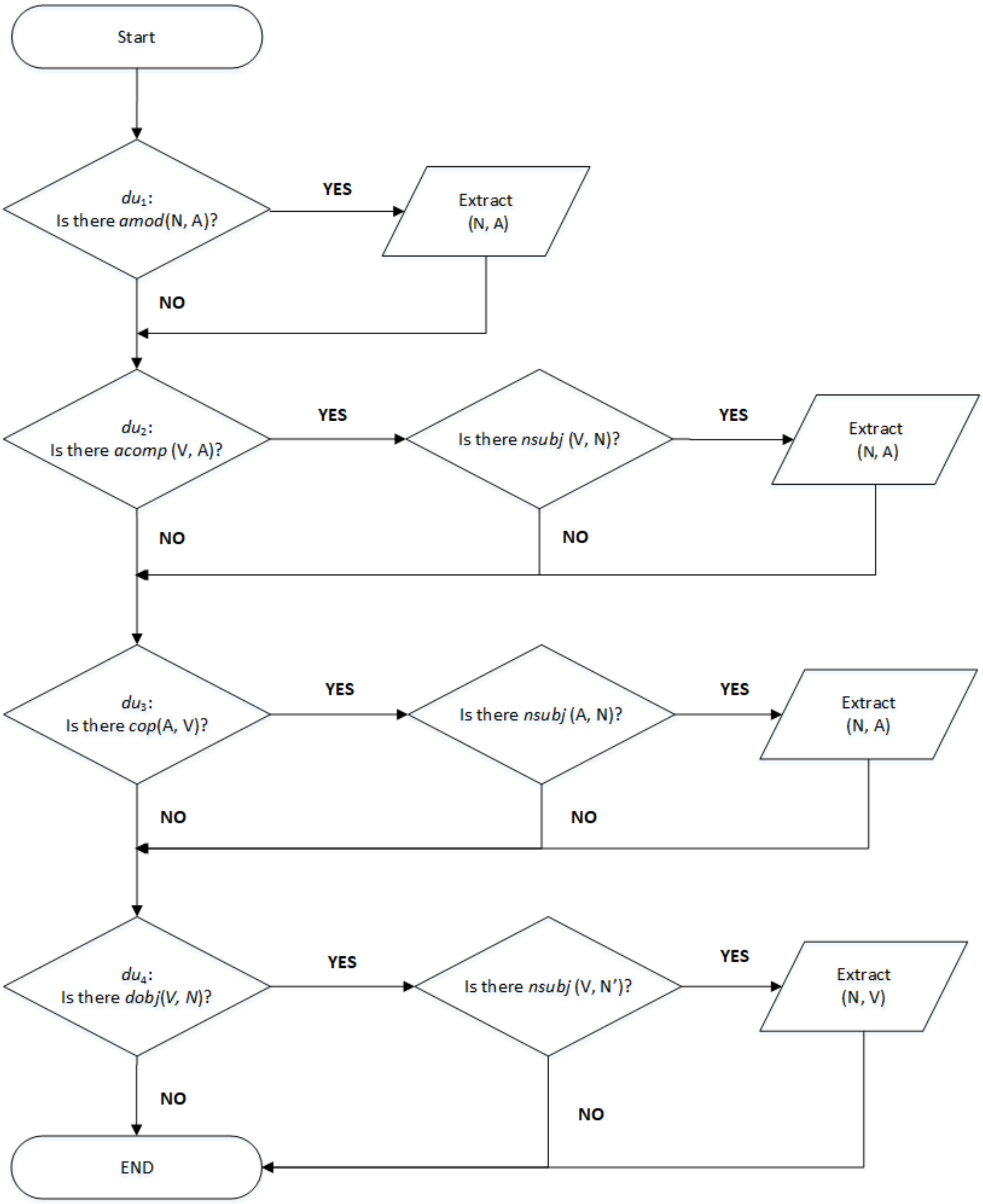


FIGURE 2.13: Rules du_1 to du_4

Compound (*compound*) exists in the list of dR, hence rules du_5 or du_6 apply; and in this example rule du_5 triggers, resulting in the following output:

$$(lens, good) + compound(lens, camera) \rightarrow \langle camera\ lens, good \rangle.$$

In this way, given a set of reviews a set of candidate phrases is extracted. Chen et al. (2014) proposed to improve this approach by pruning the candidate aspect phrases using

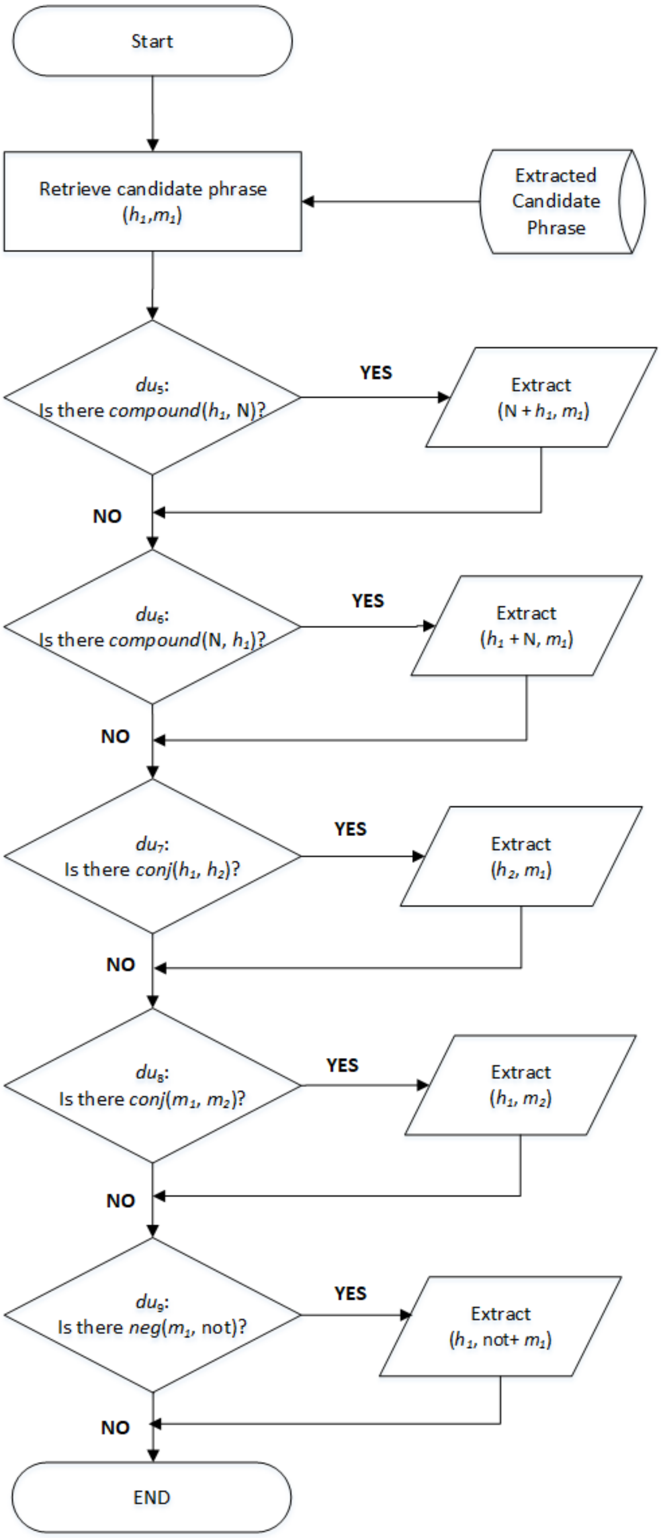


FIGURE 2.14: Rules du_5 to du_9

a frequency cut-off. To do this, for each candidate any non-noun (N) words are eliminated. Thereafter the frequency of each noun (N) and compound nouns (NN) phrase is calculated, retaining only those candidates above a frequency threshold. Evaluation results show that by pruning the candidate phrases using the frequency cutoff, precision score is significantly improved compare to not having the frequency cutoff. The proposed aspect extraction approach is applied in a recommendation algorithm that recommends electronic products to users.

Previous work applies aspect extraction algorithms proposed in the literature to extract aspects with an assumption that aspects extracted by these algorithms are meaningful to the recommendation algorithm. However, most aspect extraction algorithms proposed in the literature are evaluated on benchmark datasets and evaluation results obtained from these benchmark datasets are not guaranteed to generalise to other datasets (Holte, 1993). Therefore, it is important to evaluate the aspect extraction approach in a recommendation setting to ensure that meaningful aspects are applied in the recommendation algorithm.

2.2 Aspect Selection

NLP-based aspect extraction techniques when applied to informal text generate many erroneous aspects. Using as inspiration the field of text classification research, where feature selection approaches have been successfully used for dimensionality reduction, this thesis explores the transferability of such selection heuristics for the aspect selection task. Feature selection can be broadly categorised into either supervised and unsupervised methods.

2.2.1 Supervised

Supervised selection heuristics have been successfully employed to reduce dimensionality and achieve significant gains in accuracy for text classification (Wiratunga et al., 2004). Recent work in contextual recommender systems have also used supervised methods to evaluate the relevance of aspects given different contexts (Chen and Chen, 2014). A comparative analysis of three traditional feature selection techniques: Information Gain (IG), Mutual Information (MI) and Chi-squared Test (CHI); shows CHI to perform best on classifying contextual aspects. This performance gain by CHI is due to its ability to consider aspect and context dependencies in terms of co-occurrence frequency. However, CHI is not reliable for low-frequency terms (Yang and Pedersen, 1997), hence it will treat rarely occurring aspect terms (e.g. the aspect *aperture* in camera) as less discriminative.

One of the main challenges in using product reviews is the lack of labelled data for classification. This is because unlike with typical classification tasks where class labels are explicitly defined for each text document, product reviews labels need to be available for individual sentences, making this far more demanding. For instance, Wang et al. (2016) used users' helpfulness vote to rank the aspects by combining IG and sentiment strength. The ranking score of an aspect is determined by its absence and presence in helpfulness and helplessness reviews, and the sentiment strength expressed on the aspect. However, not every review datasets contain helpfulness votes (e.g. IMDB dataset[9], SemEval dataset[10]). In the absence of helpfulness votes, user ratings are used as class labels for classification (Vargas-Govea et al., 2011).

Miyahara and Pazzani (2000) and Billsus and Pazzani (1998) formalise a collaborative filtering recommendation problem as a classification problem. As a classification problem, their task is to predict whether a movie is liked or disliked by users. Feature selection was used to find a subset of N most informative users (liked-minded users) for making a prediction. This is accomplished by computing the expected information gain that the feature value ("like" or "dislike") of a user contributes to the classification of a set of labelled items that have been rated by the target user. When defining class labels using user ratings, they transform a numeric 6-point scale rating ranging from 0 to 1.0 (0, 0.2, 0.4, 0.6, 0.8, 1.0) to two class labels: likes and dislikes. They label the items score in the first quartile of the rating scale (0.7, which is the midpoint between 0.6 and 0.8) as items that user likes, or dislikes if the item was given a rating of less than 0.7. This approach is useful if the goal of the recommendation task is to identify aspects that are relevant in distinguishing between products that the users like and dislike. However, if the task is to identify aspects that are relevant in predicting the exact rating of a product, this approach is not recommended.

[9] https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset/

[10] http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

2.2.2 Unsupervised

The lack of labelled data and the increase of user-generated content creates a need to select useful aspects without supervision. Generally, the most popular unsupervised methods applied in review texts rely on heuristics that are informed by frequency word counts (Tsur and Rappoport, 2009). This is because a missing frequent aspect will reduce precision more than infrequent aspects. However, Wang et al. (2016) and Zha et al. (2014) argued that depending solely on frequency tends to remove important aspects that are infrequent and therefore not ideal for aspect ranking. This problem can be addressed by combining frequency counting with heuristic rules. For instance, the score of the aspect is determined by the number of times it co-occurs with an opinion word (e.g. adjective) (Eirinaki et al., 2012, Zhang et al., 2010b). Similarly, Zha et al. (2014) developed a probabilistic aspect ranking algorithm using aspect frequency as prior knowledge to infer the importance of an aspect from product reviews and then produce a ranked list of aspects based on the aspect's importance score. They evaluate the usefulness of aspect ranking in two applications: sentiment classification and text summarisation, and obtained promising results. However, it is not clear how important aspects were selected when applying important aspects in these two tasks.

Frequency counting can be a fair indicator of an aspect's utility, an alternative approach to measure relevance of an aspect is using similarity measures. Ronen et al. (2013) applied a similarity metric to measure the similarity between candidate aspects and aspects from products that were bought by the user. This is because aspects of a product which have high similarity to past purchased products' aspects are considered more relevant than those that do not. However, this approach does not work well in high priced product domains since users will not be able to provide sufficient product purchased history for similarity computation.

2.3 Chapter Summary

This chapter presented a review of the literature related to aspect-based sentiment analysis and aspect selection. To understand the application of ABSA in recommender systems, related work in this area were presented. Further, feature selection techniques are discussed in the context of recommender systems. The related works in social recommender systems as well as evaluation approaches for recommender systems will be described in the next chapter.

Chapter 3

Social Recommender Systems

This chapter presents a review of existing literature related to social recommender systems. First, commonly used social content for enhancing recommendations is discussed with a particular focus on product reviews and users' purchase preferences. The state-of-the-art recommendation methods that employ product reviews and implicit feedback are analysed. Thereafter, this chapter discusses the evaluation datasets and metrics of social recommender systems. Finally, this chapter concludes by summarising the key findings from the literature review from Chapter 2 and 3.

.

3.1 Social Content

The increasing popularity of social media allows people to share their opinions, interact with other users and enrich people's social activities with their families and friends (e.g. Twitter, Facebook etc). New forms of social content provide opportunities to enhance recommendation systems (Guy, 2015). How to incorporate these new forms of social content to improve recommendation remains an active area of research.

Making sense of user interactions is likely to reveal reasons behind user preferences. Such interactions typically manifest themselves as explicit and implicit user feedback which can be used to infer user preferences. Jawaheer et al. (2014) identified several commonly used social content types that are organised into explicit and implicit user feedback in Table 3.1.

Category	**Sources**	**Description**
Explicit	Ratings	Ratings for items are organised into a Likert Scale. The rating scale will usually from 1-star to 5-star rating.
	Social tags	Unstructured annotations in the form of short messages that describe a resource, feeling or impression.
	Microblogs	Short messages that describe, update and share users' current status and opinion.
	Product reviews	User generated content that allow users to comment on products they have purchased.
Implicit	Social relations	Online relationship between users which allows online users to share ideas and information between connected users.
	Purchase records	Users' purchase transaction history.
	User click behaviours	Users click-through log that contains a large amount of information on the items that the users have viewed and/or purchased.

Table 3.1: Explicit and Implicit User Feedback

Explicit feedback such as user ratings, social tags, microblogs and product reviews can be commonly gathered from e-commerce websites or other social media platforms such as Twitter and YouTube. These resources are rich in information, allowing users to express positive and negative opinions. While traditional collaborative filtering and content-based approaches rely on user ratings for recommendation (Koren et al., 2009, Sarwar et al., 2001), recent research has used other types of explicit feedback such as microblogs (Zhao et al., 2014) and social tags (Horsburgh et al., 2011, Zhao et al., 2008). In microblogs, short messages are used to describe, update and share users' current status and opinion. These are continuously updated allowing systems to capture users' recent purchase preferences. However, the limitation on the number of characters per post in a microblog limits expressiveness about the purchase experience. On the other hand, social tags are unstructured text annotations that are used to described users' feelings or opinions. Therefore, items that are tagged by users can directly reflect their opinions.

One major disadvantage with social tags is that it is difficult to extract interesting topics due to the dynamic nature of this informal vocabulary.

An increasing effort is being focused on incorporating knowledge from product reviews into recommendation algorithms (Aciar et al., 2007, Chen and Wang, 2013, Dong et al., 2016, Wang and Chen, 2012, Wang et al., 2013, Zhang et al., 2012). In particular, the rich information embedded in product reviews permits recommender systems to assess the quality of a product based on users' experience, and elicit users' preferences from their written reviews and ratings. The next section is dedicated to discussing the state-of-the-art techniques in review-based recommender systems that exploit product reviews for user/item representation. Implicit user feedback is then discussed in Section 3.1.2.

3.1.1 Review-based Social Recommender Systems

The use of product reviews has shown to provide better recommendation performance in collaborative filtering methods that use ratings (Chen and Wang, 2013, Ganu et al., 2013, Liu et al., 2013). The main assumption in using reviews for recommendation is that there is a correlation between the overall star rating and the aspect opinions mentioned in the reviews. Therefore, users' preferences are constructed based on the sentiments (e.g. happy, good) extracted from product ratings (e.g. 4 stars) with respect to the aspects (e.g. screen, price) which are extracted from reviews. During recommendation, users who hold similar sentiments towards product aspects are considered as similar users. For instance, Ganu et al. (2013) group similar users using a soft clustering technique based on the sentiment of aspects in the user reviews. The predicted rating of a user is the weighted average of the ratings of all other users in every cluster who have rated the product. Here, the weight is the probability with which the user belong to each cluster. This approach has shown that approaches that incorporate sentiment about aspects achieve a better prediction accuracy than those that do not. However, it assumes an equal contribution by all aspects towards the product rating; which incorrectly implies that users place equal importance to all aspects relevant to a product.

Chen and Wang (2013) and Liu et al. (2013) focus on discovering user-weighted aspect preferences from product reviews. Intuitively, if a user comments on an aspect more frequently on average then the aspect should be more important than others (Liu et al., 2013). However, this method is not able to distinguish aspects that are of equal frequency. Chen and Wang (2013) solve this problem by treating the relationship between the overall rating assigned to a product and the user's sentiments associated with the aspects as a regression problem. It is interesting to observe that predictions based on users' weighted aspect preferences are more effective for recommendation than rating-based collaborative filtering approaches.

An alternative role of reviews in recommendation is considered. Instead of eliciting users' preferences, user reviews can be used to assess the quality of products to augment product ranking. When a product is described by a set of aspects, a preference-based product ranking is possible (Smyth, 2007). Given that the product representation is based on product aspects, it is natural to consider content-based approaches to recommendation. Cosine similarity is a conventional approach in content-based recommender system research (Jannach et al., 2010, Lops et al., 2011, Pazzani and Billsus, 2007). This approach computes the similarity between query and candidate products to form a ranked list on the basis of similarity and thereby identify the k most similar candidate products for recommendation. This is based on the assumption that users are likely to look for other products (candidate products) which are similar to the product that they are currently looking at (query product). In this approach, each product is represented by a vector in an n-dimensional space, where each dimension corresponds to an aspect term from the overall vocabulary of a given corpus of product reviews. The value in each dimension is the weight of an aspect that indicates its importance, which is determined by using the TF-IDF weighting scheme. Let $\mathcal{P} = \{p_1, p_2, ..., p_N\}$ denote a set of products and $\mathcal{A} = \{a_1, a_2 ..., a_n\}$ be the set of unique aspects identified from an aspect extraction algorithm. Therefore a product p can be represented as follows:

$$p = [a_k, a_{k+1}, a_{k+2} ... a_n] \tag{3.1}$$

Here, a_k is the value for an aspect in p and n is the size of the vector. At the time of recommendation, the similarity of a candidate product (C) in a given retrieval set in terms of the target query product (Q) is measured using the standard cosine similarity metrics below (Lops et al., 2011):

$$Sim(Q, C) = \frac{\sum_{i=1}^{n} Q_i C_i}{\sqrt{\sum_{i=1}^{n} (Q_i)^2} \sqrt{\sum_{i=1}^{n} (C_i)^2}} \tag{3.2}$$

Here Q_i and C_i are the weights of the ith aspect in product Q and C respectively which are computed using the common term weighting scheme, TF-IDF (Term Frequency-Inverse Document Frequency) as follows (Christopher et al., 2008):

$$TF - IDF(a, Q, \mathcal{P}) = tf(a, Q) \times idf(a, \mathcal{P}) \tag{3.3}$$

where $\mathcal{P}$ denotes the set of products in the corpus and a is an aspect in Q. The term frequency, $tf(a, Q)$, and inverse document frequency, $idf(a, \mathcal{P})$, are given as follows:

$$tf(a, Q) = 1 + log(f_{a,Q}) \tag{3.4}$$

$$idf(a, \mathcal{P}) = log \frac{|\mathcal{P}|}{|p \in \mathcal{P} : a \in p|} \tag{3.5}$$

In Equation 3.4, $f_{a,Q}$ is the frequency of occurrence of aspect a in Q. Term frequency considers all aspects as equally important. However, aspects such as *machine* may frequently occur in the reviews of laptop products but have little importance. Therefore, the document frequency of an aspect is offset by the frequency of the aspect in the entire corpus using *idf*. The *idf* of aspect a is obtained by dividing the total number of products by the number of products that contain a and then taking the logarithm of the division.

The similarity-based approach is a simple method in product recommendation. The availability of users' sentiments in product reviews hints at an alternative recommendation approach that includes users' sentiments. BetterScore is the state-of-the-art approach that utilises users' sentiments in a content-based recommender system (Dong et al., 2016). This approach is most relevant to this work where a product case is represented as a set of product aspects that are paired with its corresponding sentiment score. Therefore, each product case, $Case(P)$, is represented as follows:

$$Case(P) = \{(a_j, Sentiment(a_j, P)) : a_j \in \mathcal{A}(P)\} \tag{3.6}$$

where the product aspects $\mathcal{A}(P)$ for a product P are all the product aspects discovered from the reviews of P. The sentiment score of aspect a_j in product P ($Sentiment(a_j, P)$) is assigned by aggregating the individual review-based sentiment scores of a_j.

During recommendation, products are ranked in a decreasing *Better* score order. The main idea of the *Better* score is that candidate products (C) that have a better sentiment score across the product aspects compared to the query case should be preferred by the users. For example, consider a user who is considering a digital camera X. One of the product aspects discovered from camera reviews is *battery* and the sentiment score of this aspect for camera X is 0.5. When selecting a camera for recommendation, all other things being equal, cameras which have a sentiment score greater than 0.5 will rank higher than camera X. Recall that the sentiment score of an aspect represents the overall opinion of users on whether the product aspect is good or bad. Therefore, it is reasonable to rank products according to how much *better* their aspects are than the query product. For a given query product, a set of products is retrieved based on k shared aspects (a). Formally, the *Better* score is defined as follows:

$$better(a_i, Q, C) = \frac{Sentiment(a_i, C) - Sentiment(a_i, Q)}{2} \tag{3.7}$$

$$Better(Q, C) = \frac{\sum_{a_i \in \mathcal{A}(Q) \cap \mathcal{A}(C)} better(a_i, Q, C)}{|\mathcal{A}(Q) \cap \mathcal{A}(C)|} \tag{3.8}$$

The starting point for computing the *Better* score is to compute a *better* score between the shared product aspects of Q and C. A *better* score that is less than 0 means that Q has a better sentiment score for a_i than C. In contrast, a positive *better* score means that C has a better sentiment score for a_i than Q. Then, an overall *Better* score is computed at the product level by aggregating the individual *better* scores for each product aspect, returning a score between -1 and +1. Instead of only using the number of shared aspects to measure similarity between products, a finer-grained similarity is derived by comparing the sentiment score differences between aspect pairs from the query and candidate product (Dong et al., 2016). The retrieval set is obtained by measuring a weighted similarity between the query and candidate products using the frequency of the aspects mentioned in the reviews as a function of similarity as follows:

$$score(q,p) = (1-\alpha) * sim(q,p) + \alpha * Better(q,p) \tag{3.9}$$

One of the limitations observed from this work was the assumption that users place equal importance to all aspects relevant to a product. Another limitation observed in this approach is that analysing users' sentiments on product aspects leads to recommendations that are of better quality (e.g. having better rating) than the query product. It has been shown that, when the recommendation is solely based on sentiment scores of a product the list of better products recommended tends to be less similar to the query product (Dong et al., 2016). This implies that the products recommended might be very different from what the user wanted. Therefore, the recommendation strategy needs to be improved in such a way that priority is given to products that are similar to and have a better quality than the query product.

Aspects may influence purchase behaviour differently and as such it becomes natural to consider that aspects may also have different levels of importance (Muhammad et al., 2015). A common method is to estimate user's weighted aspect preferences through aspect frequency counts in reviews. However, in the product domain especially with high price products such as DSLR cameras, it is unlikely that a user will provide reviews for multiple products in the same product category. This can be supported by the statistics reported in Jindal and Liu (2008) and Xie et al. (2012) where most of the reviewers provide feedback on a single product (>68% in Amazon dataset and >90% in resellerratings.com dataset). Therefore, it is not feasible to use frequency counts to estimate importance of an aspect to a user.

Apart from directly expressing opinions about aspects of a product, users also express opinions by comparing similar products with respect to their shared aspects (Ganapathibhotla and Liu, 2008). For example, a direct opinion sentence that appears in a review is "*The camera has a good lens*", and a comparative sentence is "*Camera X has a better lens than camera Y*". Here, the comparative sentence does not explicitly state that any camera's lens is good or bad. Instead, it states the relative ordering of the quality of the lens of the two cameras. Therefore, opinions in comparative sentences have been found to be useful in product recommendation because potential customers would be most interested in purchasing products that are better overall than the competition. Zhang et al. (2010a) proposed to use the comparative relations found in reviews to rank products. To do this, a weighted and directed graph for each aspect models the comparative relations found in the reviews. In each graph, a node is a product (p) and an edge represents a comparative relation between two products (e_{ij}) as shown in Figure 3.1. A comparative sentence in the review of product p_j that compares product p_j and p_i with respect to an aspect a will have an edge directed from p_i to p_j. To assign a weight to the edge, if the comparative sentence implies that p_j is better than p_i then this is considered as a positive comparative (PC). On the other hand, if the comparative sentence implies that p_j is worse than p_i then this is a negative comparative (NC). Accordingly, an edge (e_{ij}) is weighted using the ratio of PC/NC.

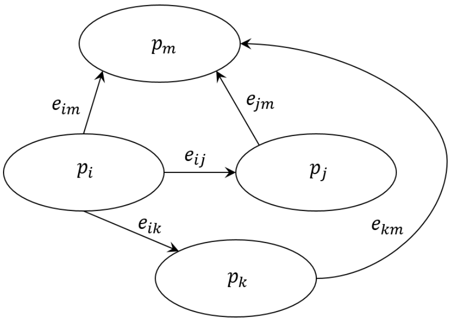


FIGURE 3.1: Directed Graph for aspect a

Recommendation based on similarity metrics and users' sentiments requires extracting product aspects and user's opinion from product reviews. An alternative approach to rank products is based on the popularity of a product. PageRank is a popular link analysis algorithm used by Google search engine to rank websites (Page et al., 1999). Web pages in the World Wide Web follow a graph-based structure of nodes (web pages) and edges (links). Links that connect web pages help search engines understand the relationship between pages. These relationships help search engines rank web pages in their search results. PageRank measures the popularity of a web page based on the probability that at a certain time, a random surfer (or a random user) will land on a web page by clicking on a hyperlink (Page et al., 1999). As a result, PageRank assigns a real number to each node in a graph with an intent that the higher PageRank score of a node, the more popular (or important) it is. Generally, the PageRank score for a given page u is computed as follows:

$$PR(u) = 1 - d + d \sum_{v \in B_u} \frac{PR(v)}{L(v)} \tag{3.10}$$

where B_u is the set of pages that link to page u and $L(v)$ is the number of outgoing links from page v. Here, d is a decay factor which represents the probability that the user stops clicking links from the page and requests a random page. A common choice for the decay factor is set at 0.85 (Ding, 2011, Gori et al., 2007).

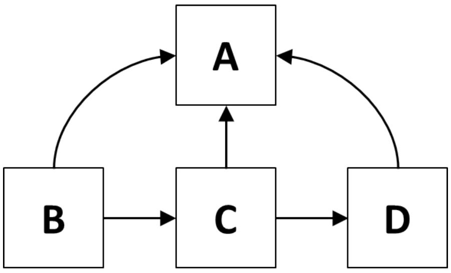


FIGURE 3.2: Webpage Graph

To illustrate the algorithm, consider a trivial example shown in Figure 3.2. In this example, there are four web pages: A, B, C and D, and the directed arrows represent hyperlinks. In the first iteration, PageRank is initialised to the same page weights for all web pages. Hence, the initial value for each page is 0.25. In the first iteration, the PageRank of page A ($PR(A)$) is calculated as follows:

$$PR(A) = (1 - 0.85) + (0.85 * (\frac{PR(B)}{2} + \frac{PR(C)}{2} + \frac{PR(D)}{1})) \tag{3.11}$$

After the completion of the first iteration, the PageRank score of page A is 0.575 (Equation 3.11). The PageRank score is computed iteratively until convergence is reached.

In the context of recommendation, PageRank is a popularity-based approach that recommends items based on their popularity in a graph-based recommender system (Ding, 2011, Gori et al., 2007, Jannach et al., 2013, Zhang et al., 2010a). The PageRank algorithm has been applied to such graph-based structures to compute an overall ranking of products with respect to a product aspects (Wang and Wang, 2014). Products which have a higher PageRank score indicate that the product receives a high number of positive feedback on aspect a. However, the classic version of this algorithm treats all edges equally and does not take into account the node weights. An extension by Zhang et al. (2010a) incorporates edge weights to compute a product ranking. Here, node weights are determined by the opinions in subjective sentences and edge weights are inferred from comparative sentences. Another related approach uses comparative sentences in community-based question answering pairs, and user reviews from multiple review websites to construct graph models (Li et al., 2011). Here, the superiority score of a product is calculated through performing graph propagation. In Jamroonsilp and Prompoon (2013), a further distinction is made between both in and out links such that the superiority of a product is not based on the products that links to it but also penalised by the number of products that the product links to. All things being equal, a product that only links to a few other products is thus preferred over one which links to several other products.

3.1.2 Recommender Systems with Implicit Feedback

A key issue with any recommendation technique is that neither user ratings nor item metadata are available in sufficient quantity. Implicit feedback aims to avoid this bottleneck by inferring user preferences from their interaction patterns instead of user's directly providing preference on product aspects. For instance, researchers have attempted to exploit social network relations for recommendations (Beilin and Yi, 2013, Yang et al., 2012b). By exploiting the fact that two socially connected users are more likely to share similar interests, recommendations can be generated by using a collaborative approach based on the social network relationship links. However, relationships in social networks are often dynamic, creating a challenge to manage user preferences.

Implicit feedback is also based on observable user interactions with the system. For instance, users' purchase records provide an indication on user preferences. In music recommendation, the number of play counts is an implicit feedback indicative of their level of interest in the song (Pacula, 2009, Parra et al., 2011, Yang et al., 2012a). Similarly in search engines, user clickthrough logs contain large amounts of information on search interests. Every click a user performs is integrated with relevance feedback methods. However, it has been argued that users' clicks can be biased (Cao et al., 2010). This is because the ranked list of results presented can unduly influence users to prefer top ranked results over those appearing further down the list.

In restaurant recommendation, Vasudevan and Chakraborti (2014) mined user's trails from a restaurant recommendation system called Entree[1] to estimate the utility of a restaurant. A user trail is a path that the user follows when searching for a product of interest. The path starts from a restaurant as an entry point, users receive a recommendation that they critique (*cheaper, creative, lively, nicer, quieter, traditional*) to look for other restaurants that suit their preferences. This cycle continues until the user stops the search. An interesting observation from this work is that they modelled the users' trails as a preference graph to estimate relative utilities of restaurants. To illustrate their approach, consider an example of the user's trails shown in Figure 3.3. Here, a user starts from r_1 and critiques to find a cheaper restaurant. The user browses r_3 and r_{10} and reach r_2 before they provide another critique *cheaper*. This trail suggests that the user prefers r_2 over r_1 as a cheaper alternative among the given recommendations. The preference graph that models the users' trails for critique *cheaper* is shown in Figure 3.4.

$$\boxed{\begin{array}{l} r_1 \xrightarrow{cheaper} r_3 \xrightarrow{browse} r_{10} \xrightarrow{browse} r_2 \xrightarrow{cheaper} \\ r_5 \xrightarrow{cheaper} r_9 \xrightarrow{nicer} r_7 \xrightarrow{browse} r_8 \xrightarrow{browse} \\ r_{23} \xrightarrow{livelier} r_{12} \end{array}}$$

Figure 3.3: User's Trails

[1] http://kdd.ics.uci.edu/databases/entree/entree.data.html

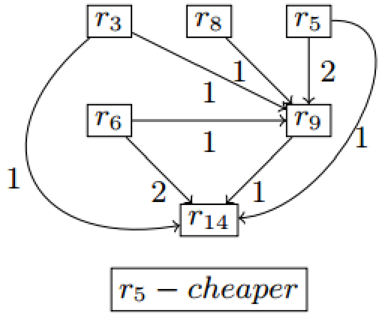


FIGURE 3.4: Users' Trails Preference Graph

In Figure 3.4, nodes are restaurants and weighted edges indicate the number of times the destination node is preferred after critiquing the source node. The approach in measuring the utility of a restaurant in this work is similar to Zhang et al. (2010a) where a PageRank algorithm was naturally adopted over the graph to identify a ranking over restaurants. The main difference in Zhang et al. (2010a) was the use of comparative sentences to model the preference graph while Vasudevan and Chakraborti (2014) exploited user trails for recommendations. One limitation observed from using the users' trails to model the preference graph is that there is no clear preference indication presented. The end point of a user trail is an indication of the restaurant that the user last saw, but it does not necessarily mean that the user is satisfied with the recommendation. In order to have a clearer picture about users' preferences, a user's transaction history can be exploited as a knowledge source for recommendation algorithms (Choi et al., 2012, Jannach et al., 2013). This form of information is very useful as it indicates clearly the type of product that the users prefers. However it requires specific access privileges (e.g. permission from an e-commerce company), which is not easily available for research purpose.

3.1.3 Combining Explicit and Implicit Feedback

Implicit feedback is abundant but it is not always knowledge-rich. For example, the fact that a consumer viewed and compared multiple products prior to a purchase may show product preferences but there is a lack of substantial evidence as to why the product was preferred. Explicit feedback quantifies a user's preference (e.g. user indicates their level of preference using user's ratings) whilst implicit feedback, which is based on the frequency of an action taken by the user, indicates confidence. A high user rating does not necessary indicate higher preference. For example, a user may watch a movie once and give a 5-star rating. However, if the user really liked the movie they could watch it more than once. A one-time event might caused by various reasons that may not be related to users' preference but a recurring event is likely to indicate a higher preference. Therefore, implicit feedback is suitable as a confidence measure for a preference model. Pioneering work on combining explicit and implicit feedback by Koren (2010) used ratings and pseudoimplicit ratings. Here, a pseudoimplicit rating is a binary value that represents whether a user has rated a movie or not. The results from this work show that combining both types of feedback yields better performance as compared to solely using explicit feedback. Similar conclusions can be drawn from more recent work where different sources of explicit and implicit feedback such as explicit user preferences and ratings, user's listening behaviour and user's rating behaviour (Li and Chen, 2016, Moling et al., 2012) are combined. This is an important finding as it sheds light on current state-of-the-art approaches that rely on hybrid methods.

3.2 Evaluation

Recommender systems evaluation requires a dataset to apply the proposed algorithms and appropriate metrics to measure recommendation performance. This section discuss the datasets and the evaluation metrics that are available for evaluating recommendation performance.

3.2.1 Datasets

This thesis focuses on datasets used in evaluating review-based recommendation algorithms and datasets which are widely used in evaluating aspect extraction algorithm.

3.2.1.1 Review Datasets

At the time of starting this work, the availability of public review datasets was limited. Some of the notable datasets that were used to evaluate review-based recommender systems are the following:

- Yelp Dataset[2]: This dataset contains 5,996,996 reviews of 188,593 businesses (includes restaurants, nightlife, local services and delivery) from 1,518,169 users. The businesses are rated on a 1 - 5 star scale.
- IMDB Dataset[3]: This dataset contains 50,000 movie reviews with each movie containing no more than 30 reviews. Ratings on IMDB are given on a 1 - 10 star scale.
- Amazon Dataset[4]: This dataset contains 34,686,770 reviews from 2,441,053 products. Each product includes product and user information, ratings, and text reviews. Products are rated on a 1 - 5 star scale.

These datasets do not provide users' implicit feedback, such as user purchase behaviours that are implicitly captured for instance by click-through and customer purchase information. Therefore, these datasets are not suitable to evaluate social recommender systems that capitalise on implicit feedback to generate recommendations.

3.2.1.2 Aspect Extraction Datasets

To evaluate an aspect extraction approach, labelled datasets are required. The most widely used annotated datasets that are publicly available are the following:

- Hu and Liu Dataset[5]: This dataset consists of 5 electronic products reviews crawled from Amazon.com and Cnet.com (2 digital cameras, 1 cellular phone, 1 mp3 player and 1 dvd player).

[2] https://www.yelp.com/dataset
[3] https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset/home
[4] https://snap.stanford.edu/data/web-Amazon.html
[5] https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

- SemEval Dataset[6]: This corpus consists of two domain-specific datasets for laptops and restaurants. For each domain, there are approximately 3,000 sentences for training and 800 sentences for testing.

In general, these datasets are used to measure the precision and recall of a proposed aspect extraction algorithm. However, this thesis aims to propose an aspect extraction approach that improves recommendation performance (e.g. recommend '*better*' products) and also to explore how different aspect extraction algorithms affect the end recommendation performance through a comparative study. Algorithms which achieved the highest precision score in these datasets may not necessary translate to a better recommendation performance. This is because data characteristics such as dimensionality and noise level affect evaluation performance, and approaches which work best in an isolated manner do not necessarily result in a better recommendation when compared to simpler but more robust approaches (Japkowicz and Shah, 2011). Therefore, in order to evaluate the effect of an aspect extraction algorithm in recommendation performance, the aspect extraction algorithm needs to be evaluated in a recommendation setting on a suitable dataset.

3.2.2 Evaluation Metrics

There are several properties of recommender system that can be evaluated, for example accuracy, novelty, diversity and serendipity (Herlocker et al., 2004, Kaminskas and Bridge, 2017). Measuring these properties can be achieved through different experimental methodologies; qualitative online user studies and quantitative empirical studies (Ricci et al., 2011). Whilst a qualitative study is preferred in practice they require access to a realistic e-commerce setup, which makes it prohibitive for a small-scale research projects. Instead this work focuses on the general category of empirical quantitative methods. Therefore, the main challenge is in the formulation of appropriate evaluation metrics that capture recommendation performance using an appropriate dataset.

[6] http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools

Specifically, with empirical evaluation it is normal to split data into training and test sets. Then for each user in the test set the system can generate recommendations and measure the effectiveness of the system's performance using evaluation metrics. Broadly, metrics can be categorised into two major classes: accuracy and non-accuracy metrics.

3.2.2.1 Accuracy Metrics

The most common task in recommender systems is to present a ranked list to identify the top-N recommendations. An alternative task involves rating estimation whereby the system predicts ratings for unrated items of a user. Most commonly the RMSE (Root Mean Square Error) and MAE (Mean Average Error) are adopted with the goal of achieving similar product rating (e.g. on a Likert scale) to that of a gold standard with smallest error. Rating prediction is most appropriate for scenarios where an accurate prediction of the ratings for all products is required. When the rating prediction error rate is not a concern, recommendation performance can be evaluated using a ranking prediction task.

Top-N recommendation is a ranking prediction task that is commonly used in e-commerce websites (e.g. Amazon, Netflix, TripAdvisor) to provide a list of N recommended products that is likely to be of relevance to the user. Thereafter the user can view or purchase one or more of the suggested products. A top-N recommendation task can be viewed as a ranking task (Schröder et al., 2011). The aim of the system is to rank relevant products at the top of the list. Here, basic information retrieval metrics such as precision and recall are commonly used to establish relevance of the ranked list. Precision measures the ratio of the retrieved products being relevant and recall measures the ratio of all relevant products contained in the retrieval set. In information retrieval (IR), both precision and recall are combined in metrics (e.g. F-measure) that can be parameterised to favour one over the other (Herlocker et al., 2004).

In recommender systems, metrics such as Normalised Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) are commonly used to evaluate ranking algorithms. NDCG measures the usefulness (or gain) of an item based on the position of a product in the ranked list. There are two assumptions in this metric (Christopher et al., 2008):

- highly relevant products are more useful to the user than marginally relevant products.
- the lower the ranked position of a relevant product, the less useful to the user since it is less likely to be seen by the user.

As a measure of usefulness, users give a non-zero relevance value to each item (e.g. between 0 to 3). The gain is accumulated from the first top product in the ranked list and discounted when relevant products appear in the lower ranked list. Formally, NDCG at p is the total gain accumulated at product rank position p:

$$NDCG_p = \frac{DCG_p}{IDCG_p} \tag{3.12}$$

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{3.13}$$

Where DCG is Discounted Cumulative Gain and IDCG is Ideal Discounted Cumulative Gain, which is the DCG of an ideal ranked list that is sorted according to a user's relevance value. IDCG is computed using DCG in Equation 3.13. Here, rel_i is the relevance value of the product to the user at position i. Similar to NDCG is MAP which is computed by calculating the average precision at every recall level. Then, an arithmetic mean of the average precision of all queries is calculated to get the final mean average precision as defined as follows:

$$MAP = \frac{1}{N} \sum_{j=1}^{N} \frac{1}{|Q_j|} \sum_{k=1}^{|Q_j|} Precision(k) \tag{3.14}$$

where Q_j is the number of relevant products for query j, N is the number of queries and $Precision(k)$ is precision at kth relevant item in the retrieval set. With both metrics, the order of the relevant items in the recommended list influences the final score. This means that relevant items placed at lower ranks will accrue penalties. However, in NDCG the user must provide the relevance score of an item. When this is not available, MAP is a better choice of metric to use.

A ranking task sees the order of the items as important. Correlation metrics such as Kendall's Tau and Spearman correlation coefficient are often used to measure ranked order alignment to a gold standard. The main advantage of these metrics is their simplicity and statistical basis. However, one major pitfall for these metrics is that the result will become less meaningful when the list contains many identical scores or ratings. This is because the evaluation might state that the system is less efficient due to the need to break ties effectively.

3.2.2.2 Non-Acccuracy Metrics

There are several properties of system usability that can be evaluated besides accuracy. These include diversity, novelty and serendipity of recommendations. Diversity reduces redundancy under the assumption that presenting too many similar items will be less useful, calling instead for more varied choices (Kaminskas and Bridge, 2017). The notion of diversity has association with novelty and serendipity. Systems that evaluate based on these properties aim to provide new (novel) and surprising (serendipity) items to the users (Herlocker et al., 2004). However, a novelty bias can run the risk of irrelevant items being elevated. The main challenge for evaluation with non-accuracy metrics is the lack of accessibility to user preferences. Furthermore, novelty, diversity and serendipity may come at the expense of accuracy.

There are other evaluation strategies and metric proposed in the literature that attempt to replicate real-world user evaluations. In Horsburgh et al. (2011), recommendation performance is measured based on the strength of association between liking and listening to a track. Recommendation algorithms with highest association scores are suggestive of high quality recommendations. This approach is interesting as it gathers global agreement on the association between a pair of tracks. However, it is less suited to an e-commerce domain particularly given the high priced products (e.g. DSLR cameras, TV). Essentially it is unlikely that there are many users who will buy different versions of the same product making it hard to establish associations. However, this approach is useful when recommending accessories for high priced products. For instance, it is more likely to establish associations between a SD memory card and a purchased camera.

The position of a recommended product in a ranked list can be used to construct evaluation strategies that are not based on predicting user preferences. Dong et al. (2013) use Rank Improvement (RI) to evaluate recommendation performance in terms of whether the recommended product has a higher overall rating than the query product. To do this, the average gain in rank position of recommended products over the left-out query product is computed relative to a benchmark ranking. Ideally, the recommended products should have a higher user ratings than the query product.

Formally, the RI of a ranked list of recommended products is defined as follows:

$$RankImprovement(RI) = \frac{\sum_{i=1}^{n} benchmark(p_q) - benchmark(p_i)}{n * |\mathcal{P} - 1|} \tag{3.15}$$

Here, *benchmark* returns the position of a recommended product p_i on the benchmark ranking, n is the number of recommended products and $\mathcal{P}$ is the number of products in the benchmark ranking. In this metric, the greater the rank gain of the recommended product over the query product (p_q) the better the recommendation. Suppose the query product is ranked 40th on the benchmark ranking of 81 unique products, and the recommended product is ranked 20th on the benchmark ranking list, then the recommended product will have a relative rank improvement of 25%. This approach is interesting because the aim of the evaluation is to measure the system's ability in recommending a better product relative to the query product, therefore constructing a user profile is not needed. However, it is important to consider the similarity between a query and candidate product. This is to avoid having higher ranked products that are not similar to the product that the user wanted.

3.3 Statistical Tests

The purpose of statistical tests is to compare the overall performance of different algorithms. In general, there are two categories of statistical tests: parametric tests such as paired t-test and ANOVA (Analysis of Variance) (Fisher, 1956) and non-parametric tests such as Wilcoxon signed-rank test (Wilcoxon, 1945) and Friedman test (Friedman, 1940). Parametric tests make the strong assumption that the samples are drawn from normal distributions. Furthermore, parametric tests also assume that the correlated samples have equal variances. This assumption is the most difficult to ascertain and has potential implications to post-hoc tests when comparing multiple algorithms on multiple domains (Japkowicz and Shah, 2011). In contrast, non-parametric approaches do not make such assumptions. Demšar (2006) and Shani and Gunawardana (2011) recommends non-parametric tests for comparing multiple algorithms as these tests do not assume normal distribution or homogeneity of variance. Specifically, when comparing two algorithms for one or more datasets, the Wilcoxon signed-rank test is recommended; however, when comparing multiple algorithms over multiple datasets the Friedman test should be used.

When a statistical test comparing multiple algorithms shows that there is a performance difference between the algorithms (the null hypothesis is rejected), post-hoc tests are performed to identify which algorithm differs in performance. Post-hoc tests are categorised into parametric and non-parametric tests. Some of the examples of the parametric post-hoc tests are Tukey test (Tukey, 1949) and Bonferroni-Dunn test (Dunn, 1961). The Tukey test makes pairwise comparisons of algorithms' performance to find out whether their difference is significant. The Bonferroni–Dunn test is similar to the Bonferroni test[7] except that the significance level is divided by the number of comparisons made (Demšar, 2006, Japkowicz and Shah, 2011). For non-parametric tests, the Bonferroni-Dunn test can be used after a Friedman test when comparing multiple algorithms over multiple datasets. An alternative to this test is the Nemenyi test (Nemenyi, 1962) which decides whether the rank differences obtained as a result of the Friedman test are significant. A comparative study by Demšar (2006) has shown that for non-parametric post-hoc tests,

[7] The Bonferroni test uses a Bonferroni correction for multiple comparisons

the Bonferroni-Dunn test is a statistically more powerful post-hoc test than Nemenyi test.

3.4 Conclusion from the Literature

Social recommender systems harness knowledge from social content to generate better recommendation rankings. A literature survey suggests that review-based recommender systems tend to focus on using explicit feedback - product reviews, to generate recommendation. However, user interactions are a key indicator of their preferences which manifest as explicit and implicit user feedback. This thesis targets two social knowledge sources: product reviews and users' purchase preferences. User opinions captured within product reviews are a rich source of information. This is because they express not just general sentiment about a purchased product but importantly holds clues as to their reasoning with respect to specific product aspects. Recommender systems can therefore mine useful knowledge and exploit the distribution of sentiment over these aspects to improve product rankings. Similarly, users' purchased preferences provide strong evidence of users' preferences. The works in this thesis focuses on integrating both knowledge sources to model sentiment over a set of product aspects, which is expected to be more effective in product rankings (Chapter 6).

As discussed in Chapter 2, text from social media platforms is characterised by a diverse vocabulary and the presence of ambiguities. Therefore, a high performance aspect-based sentiment analysis algorithm is needed to allow social recommender systems to be used to their full potential. There are two main approaches for aspect extraction: supervised and unsupervised. A supervised approach is least favoured due to the challenges in obtaining reliable ground truth data to evaluate the performance of new algorithms. The most popular unsupervised approach extracts aspects that are found to be frequent nouns and noun phrases and may inadvertently remove infrequent yet important aspects. To overcome this limitation, dependency relation-based approaches have been adopted in the literature. However, the application of irrelevant dependencies can lead to the erroneous extraction of aspects, which will invariably have a detrimental effect on recommendation performance. Another limitation observed in previous work is that they assume all the aspects extracted using the aspect extraction algorithm proposed in the literature (e.g. frequent noun approach and dependency-based approach) are meaningful to the recommendation algorithm. However, it is important to evaluate the aspect extraction approach in a recommendation setting to ensure that meaningful aspects are applied in the recommendation algorithm. Based on these observations, this thesis proposed an informed aspect extraction approach and evaluate its performance against frequent noun approach and dependency-based aspect extraction method in a recommendation setting to analyse how these baselines affect recommendation performance.

A further approach to enhance the use of aspect-based sentiment analysis in recommendation is aspect selection. This is because social recommender systems performance rely heavily on the accuracy of the aspect extraction task. Therefore, aspect extraction and further perform selection of useful aspects are crucial. Feature selection techniques that are commonly used in text classification can be useful in this task. Supervised selection has shown to achieve significant gain in accuracy for text classification. However, survey of the recommender system literature suggests that discriminative selection heuristics such as information gain and Chi-squared test from text classification although very relevant have not been fully explored in the context of aspect selection. Therefore, techniques that are suitable in this context will be explored in Chapter 7.

To evaluate a ranked list of N products, the most common accuracy metrics are MAP and NDCG. However, NDCG requires user-provided relevance scores, which makes it less accessible; therefore MAP provides a better alternative for evaluation purposes in this thesis. Besides accuracy metrics, non-accuracy metrics such as Rank Improvement (RI) are also suitable to evaluate product ranking when user profiles are not available because it evaluates recommendation performance in terms of whether a recommended product is *better* than the product that the user is looking at. Since the evaluation datasets used in this work do not contain user profiles, MAP and Rank Improvement are used instead to evaluate product rankings.

3.5 Chapter Summary

This chapter presented a review of the literature related to the work in this thesis. Opportunity and challenges of social content used in social recommender systems were discussed. A more detailed discussion on review-based social recommender systems and implicit feedback are presented as these are closely related and motivate the work presented in this thesis. Further, datasets, significant test and evaluation metrics used by researchers to evaluate recommender systems are reviewed. Instead of focusing only on accuracy metrics, the discussion is extended to evaluation strategies that are beyond accuracy metrics. Finally, this chapter concluded with a conclusion from the literature survey.

Chapter 4

Background

This chapter presents the baseline algorithms that are used in this research to evaluate the product recommendation approaches. This is followed by a presentation of the evaluation datasets, evaluation methodology and performance metrics applied in this research.

4.1 Aspect Extraction Algorithms

Social recommender systems that analyse product reviews for recommendation use methods adapted from the domain of aspect extraction. There are three main approaches for aspect extraction: the frequent noun approach, the dependency relations model and supervised learning. Supervised approaches require annotated training data to train new algorithms. However, annotated data is not always readily available for social media text. Therefore, in the literature, the most common approaches to extract aspects from real-world datasets in recommender systems research are the frequent noun approach and dependency-based approaches. The common characteristic for these approaches is that they combine shallow NLP (Natural Language Processing) techniques and manually crafted rules to extract aspects. In this thesis, the frequent noun approach and dependency relation models are compared to evaluate on their effects on recommendation performance.

4.1.1 Frequent Noun Approach

In aspect-based sentiment analysis research, frequent noun approach is commonly used as a baseline to evaluate the precision and recall of an aspect extraction algorithm (Moghaddam and Ester, 2012, Popescu and Etzioni, 2007, Poria et al., 2014, Qiu et al., 2011). In the context of social recommender systems, the frequent noun approach is also used to extract aspects for product representation. Specifically, Dong et al. (2016) extract product aspects using a combination of shallow NLP and frequent noun approach to build a product representation. This approach is applied in Dong et al. (2016), which is the work that is closely related to the work presented in this thesis. Therefore, their aspect extraction approach as described in Chapter 2, Section 2.1.5 is applied in this thesis as the first baseline approach.

4.1.2 Dependency Relation Rules

Another aspect extraction approach that is applied in recommender systems is the dependency relation rules. Most previous works have not provided the list of dependency relations for extracting aspects for product recommendation except for Chen et al. (2014). Therefore, in this thesis, the dependency-based approach proposed by Chen et al. (2014) (as described in Section 2.1.5) is used as the second baseline approach to extract aspects. Here, the frequency threshold is set to 2 which is commonly adopted in the literature (Hu and Liu, 2004, Qiu et al., 2011).

4.1.3 SenticNet Aspect Parser

In section 2.1.2, a state-of-the art aspect extraction algorithm, SenticNet aspect parser (Poria et al., 2014) was introduced. The dependency-based approach proposed by Chen et al. (2014) in Section 4.1.2 extract aspects using a set of predefined dependency relation rules without relying on external knowledge source. However, SenticNet aspect parser capitalises on SenticNet sentiment lexicon and dependency relation rules similar to Qiu et al. (2011) to extract aspects. Furthermore, SenticNet aspect parser uses the dependency relation *advmod*, *advcl*, *xcomp*, *cc*, and *prep* which are not used in Chen et al. (2014). The evaluation results show that SenticNet aspect parser outperformed the frequency noun approach by Hu and Liu (2004) and Popescu and Etzioni (2007) as well as dependency propagation approach by Qiu et al. (2011). However, at the time of this thesis, application of this approach in recommender system was not observed. Therefore, this approach was selected as the third baseline approach.

4.2 Social Recommender Systems: Baseline Algorithms

Three benchmark recommendation algorithms can be used as baselines: Cosine similarity retrieval, BetterScore and PageRank. They are detailed in this section.

4.2.1 Similarity-based Recommendation

Recommender systems which capitalise on textual reviews perform a form of content-based recommendation. Cosine similarity is a standard baseline to evaluate content-based recommendation algorithms (Jannach et al., 2010, Lops et al., 2011, Pazzani and Billsus, 2007). Product aspects help describe content, and when given a query's content (in the form of aspects) each candidate product's aspect value can be compared with that of the query product. In this thesis, the cosine similarity-based approach discussed in Chapter 3, Section 3.1.1 will be used as the first baseline approach to evaluate the recommendation algorithm.

4.2.2 Sentiment-Enhanced Recommendation - BetterScore

The state-of-the-art approach that utilises users' sentiments in a content-based recommender system is the Better score (Dong et al., 2016). One major limitation observed in this approach is that it assumes that users place equal importance to all aspects relevant to a product. This thesis aims to address this weakness by inferring aspect importance from preference relations generated over product view-purchased relations. Therefore, the Better score is the second baseline used to evaluate the proposed recommendation algorithms.

4.2.3 PageRank

PageRank algorithm recommends items based on their popularity in a graph-based structure. Previous research exploit this relationship to gauge popularity of an item, whereby a PageRank score is computed by evaluating quality and quantity of links to a node (Chen et al., 2014, Ding, 2011, Wang and Wang, 2014). The most popular item in the graph will have the highest PageRank score. Several studies indicate that product popularity is a powerful form of feedback that influence users' purchase decision (Celma and Cano, 2008, Salganik et al., 2006, Zhu et al., 2012). Therefore, the PageRank approach is used as the third baseline. In order to use PageRank as the baseline, the PageRank algorithm is adapted to suit this work. The preference relation between viewed and purchased products discussed in Chapter 1 also follows a graph-based structure of nodes (products) and edges (preference relations). Therefore, an overall preference score is computed for each product included in the preference graph by applying the PageRank equation described in Chapter 3, Section 3.1.1.

4.3 Datasets and Statistics

The preference relation between products is an important source of knowledge in recommendation. However, at the time of this thesis, no public dataset containing this information was available. Therefore, this information was collected from Amazon during April 2014 and November 2014. In particular, data from seven different product categories were collected: DSLR cameras, Laptops, Tablets, Phones, Printers, Mp3Players and TV. This data includes information about the product (e.g. product name, price, date of product release), their reviews, user ratings, best seller rank and the list of products that other consumers bought after viewing a product. Since this thesis is not focusing on the cold-start problem, newer products and those without many user reviews are also removed. The 1st of January 2008 and a threshold of 10 reviews were used as the pruning factor for products. Further, it appeared that some of the purchased products were not in the same product category as the viewed product (e.g. memory cards in the DSLR camera dataset). This is because users may have decided to buy camera accessories instead of a new camera after browsing the camera products. Therefore, any purchased products that are not in the same category of the viewed product were removed as well.

Table 4.1 shows the descriptive statistics of the seven datasets used in the experiments. Here, it can be observed that Tablets has the highest number of products (122 products) with 15,007 reviews. In contrast, there are only 3,734 reviews for 121 products in Laptops which also suggests that Laptops have the lowest number of reviews per products. Although DSLR has only 56 products with 6206 reviews, it can be observed that reviews in DSLR are relatively lengthy in size with an average of 12 sentences and 200 words per review. This makes DSLR the largest dataset in the collection. The mean of the user ratings (μ) for the datasets are between 3.5 to 4.6 with standard deviations of between 0.2 to 0.7. Specifically, Mp3 has the highest standard deviation which shows that this dataset have a wider range of user ratings values that lies between 2.2 to 5.0. The number of product preference pairs for Phones and DSLR is low compared to other datasets which have more than 50 preference pairs. Specifically, Laptops has 574 product preference pairs which is the highest among all the datasets.

Descriptions	**DSLR**	**Laptops**	**Tablets**	**Phones**	**Printers**	**Mp3**	**TV**
No. of products	56	121	122	51	82	55	52
No. of reviews	6,206	3,734	15,007	2,595	11,442	4,690	5,860
Average no. of sentences	12	6	5	5	4	6	4
Average no. of words per sentence	200	102	76	80	57	81	50
Ratings, Mean (μ)	4.6	3.9	4.2	4.3	4.0	3.5	4.2
Ratings, Standard Deviation (σ)	0.2	0.4	0.5	0.3	0.3	0.7	0.3
No. of product preference pairs	48	574	212	40	110	53	77

Table 4.1: Descriptive Statistics of Amazon Dataset

4.4 Evaluation Methodology and Metrics

In this thesis product recommendation is viewed as a ranking problem. Due to a majority of the evaluation datasets (e.g. DSLR, Phones, Mp3 and TV) being relatively small and of some of them having a class imbalance problem. Splitting the dataset into training, validation and test set is not ideal since this will lead to each split not having an equal representation of products with low (rating 1 to 2), neutral (rating 3) and high (rating 4 to 5). To resolve this issue, the dataset is split into training and test set. A stratified k-fold cross validation is used to generate the training and test sets to evaluate the recommender system performance as shown in Figure 4.1. The training set is used to learn aspect weights and compute aspect ranking. Whilst the performance of the algorithms is evaluated on the test set, using the standard leave-one-out methodology, where the system recommendations for each query product in the test set were compared.

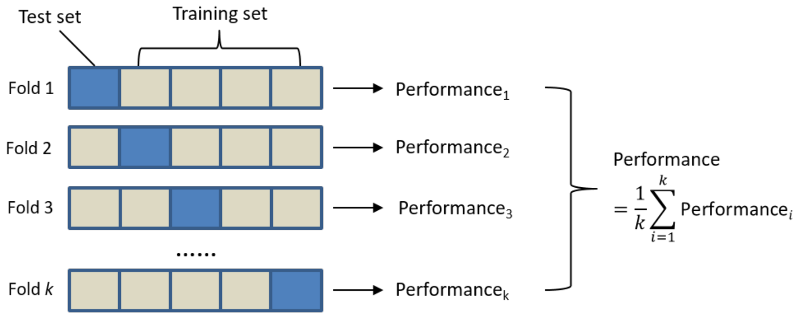


FIGURE 4.1: k-fold cross validation

To simulate a real-world e-commerce recommendation scenario, a leave-one-out methodology as described by Smyth (2007) was used. Many e-commerce websites (e.g. Amazon, eBay etc) allow users to browse or navigate through their product catalogue. When a user clicks on a product, a list of recommended products is displayed together with the details of the product that the user clicked on. Here, the query product represents the product the user clicked on. Retrieved products represent a list of products that are presented to the user. As illustrated in Figure 4.2 the evaluation methodology is focused on the user viewed "Query Product" (here product 3), followed by the steps of similarity-based retrieval and re-ranking.

For every query product, a list of m products that are most similar to the query product is retrieved using a standard cosine similarity (Equation 3.2), ranked in a decreasing order of similarity (Step 1). The retrieval set is then re-ranked using the proposed approaches (Step 2). In this way, recommended products are similar and *better* than the query product. The size of the ranked list of products depends on the size of the dataset. For example, the Phones dataset contains 51 products. After splitting the dataset into

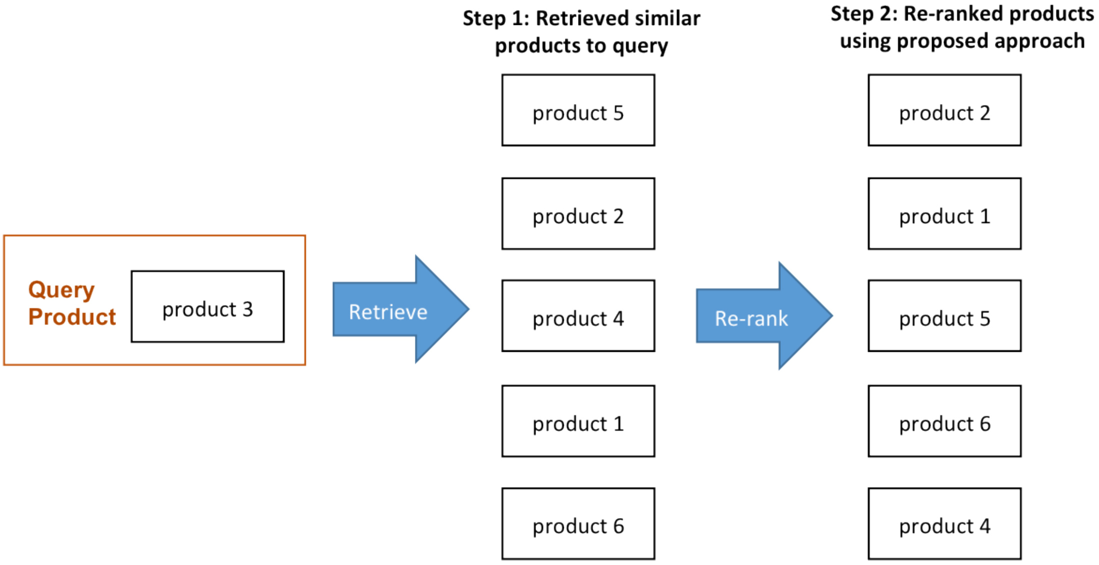


Figure 4.2: Leave-one-out Methodology in Test Set

training and test set, there are 35 products for training and 16 products for testing. Therefore, the number of products retrieved for each query product is never more than 15 (the last one being the target query product). The recommendation performance of an algorithm is estimated using the average of the results of k folds, as illustrated in Figure 4.1. The multiple algorithms results obtained from the seven datasets were tested for statistical significance using a non-parametric significance test, the Friedman test, followed by Dunn's pairwise post-hoc test with Bonferroni correction to establish where the difference was located. A non-parametric test was chosen because firstly, it does not assume a normal distribution and secondly, it does not assume the homogeneity of variance which is the most difficult to ascertain and has potential implications to post-hoc tests (Demšar, 2006, Japkowicz and Shah, 2011). Furthermore, research has shown that Bonferroni-Dunn test is statistically more powerful post-hoc test than its most common alternative, the Nemenyi test (Demšar, 2006).

4.4.1 Recommendation Tasks and Ground Truths

Two experiments were performed to measure the recommendation performance of the proposed approach. In the first experiment, the ranked lists of recommended products are evaluated by measuring the quality of the ordering among relevant products using MAP as the evaluation metric. In the evaluation datasets, the recommended products given by Amazon were not available. Therefore, the ground truth was generated using Amazon's overall product ratings as an independent objective measure of product quality to evaluate the proposed approach. For each query product in the test set, the ground truth is in the form of $(p_q, better)$ where p_q is a query product and *better* is a list that consists of the corresponding top n candidate products that are similar to and have a higher overall user rating $(better)$ than p_q. In the case where there is a tie in the product rating, the candidate product is considered *better* than the query product if it has a higher number of comments than the query product. Query products that have no '*better*' products (i.e. products which are at the top of their ranking) are not used in the evaluation. However, the first experiment does not estimate the degree in which the recommended product is *better* than the query product. Therefore, a second experiment was performed to evaluate recommendation in terms of the degree to which the recommended product is *better* than the query product.

In the second experiment, the top-n recommendations using the proposed approach for each query product were generated. This experiment estimates the degree to which the recommended product is '*better*' than the query product using Rank Improvement (RI) as the evaluation metric. The average gain in rank position of recommended products over the left-out query product is computed relative to a benchmark ranking, which is generated according to Amazon's overall user ratings of products for each test set. In the case where there is a tie in the product rating, the product which has a higher number of comments is ranked higher. In both experiments, n is set to 3 because products that are ranked at the top are likely to get users' attention or users' clicks (Christopher et al., 2008). Therefore, it is important that the products that are ranked at the top are better than the query product.

4.4.2 Recommendation Performance

Each proposed approach outputs a ranked list of products. The recommendation performance was evaluated using two evaluation metrics:

- **Mean Average Precision (MAP)** : Evaluates the recommendation performance based on the average precision across N queries. Here N is set to 15 due to the limitation on the number of products in the dataset as discussed in Section 4.4 (page 74 and 75). In this work, the aim of MAP (see Equation 3.14) is to evaluate the recommendation performance by considering the rank position of the *better* products in the recommended list such that the higher the *better* products are ranked, the higher the MAP value.

- **Rank Improvement (RI)** : Evaluates the recommendation performance based on the average gain in rank position of recommended products over the left-out query product. Ideally, the recommended products should have a higher rank than the query product (see Equation 3.15).

In an ideal case, the best performing algorithm should achieve the best result in both MAP and RI. However, a lower RI is more damaging as users are likely to feel disappointed with the recommender system if they were recommended with products that have lower ratings than the ones they are looking at. Therefore, in this thesis RI is valued more than MAP. For instance, when the results are mixed (e.g. a high MAP but a low (or negative) RI or vice versa), results for RI will be prioritised when evaluating recommendation performance.

4.5 Chapter Summary

In this chapter, the baseline algorithms for aspect extraction and recommender systems were presented with justifications on why each baseline was chosen. For aspect extraction, the three baseline approaches that include the frequent noun and dependency-based approaches were described. Then, for the recommendation task, three different baselines that are closely related to this work: Cosine similarity-based recommendation, Better score, and PageRank were also discussed. Finally, the details of the evaluation datasets, methodology and metrics that were used to evaluate the approaches presented in this thesis were presented.

Chapter 5

Dependency Rule-based Aspect Extraction and Sentiment Scoring

Aspect extraction is the first step in generating product representation prior to product ranking. This chapter describes the first contribution of this thesis which is an aspect extraction approach that extracts product aspects from reviews by combining the dependency relations and frequent noun approaches. This chapter starts with a description on the best practices in extracting aspects from dependency representations generated by the Stanford CoreNLP parser. The importance of the best practices is also discussed. As discussed in Chapter 2, previous work selects a subset of the dependency relation rules without providing information on how the rules were chosen (Moghaddam and Ester, 2012, Poria et al., 2014, Qiu et al., 2011). It is important to have this information in order to select relevant dependency rules as the irrelevant rules can result in erroneous aspects being extracted. Therefore, the design of the aspect extraction approach is discussed with a particular focus on how a subset of dependency relations is selected using POS patterns and sentiment knowledge and the importance of combining aspect pruning heuristics to improve the aspect extraction process. The extracted aspects are then used for product representation. Furthermore, a survey of the literature suggests that previous works assume that aspects extracted by the aspect extraction algorithm from the literature (e.g. frequent noun approach and dependency-based approach) are meaningful to the recommendation algorithm. Therefore, this chapter concludes with the evaluation of aspect extraction approaches in a recommendation setting. In particular, the performance of the proposed approach is compared against a frequent noun baseline approach and the existing state-of-the-art dependency-based methods discussed in Chapter 4, Section 4.1.

5.1 Extracting Aspects from Dependency Representation

Text pre-processing is a key component of many NLP tasks, including aspect extraction. It usually involves a number of small subtasks that incrementally transform raw text into a format that is suitable for vectorised representation. The use of dependency rules to extract aspects requires pre-processing on the dependency representation generated by the Stanford CoreNLP before it can be used to determine the sentiment that describes the aspects.

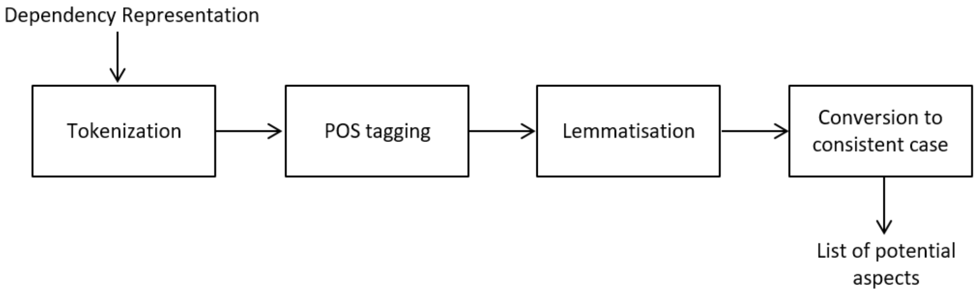
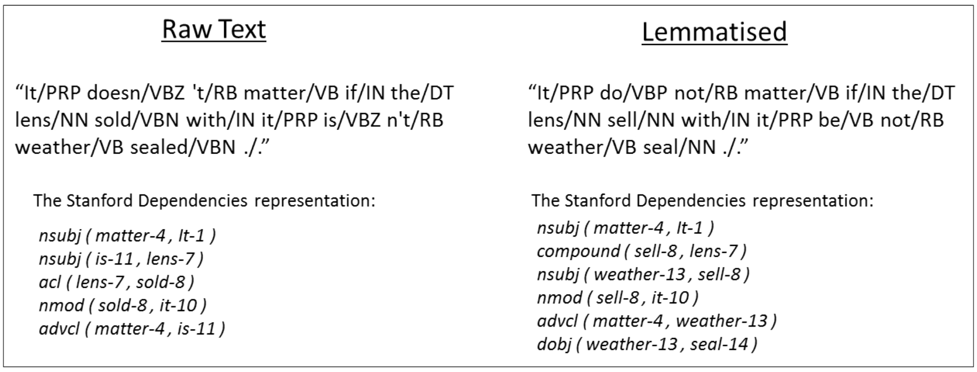

Dependency Representation
Tokenization
POS tagging
Lemmatisation
Conversion to consistent case
List of potential aspects

FIGURE 5.1: Dependency Representation Pre-processing Steps

Figure 5.1 shows the five text pre-processing steps in this research. First, aspects from each dependency representation of a sentence are extracted. Second, the resultant aspects are tagged with their respective POS (Part-of-Speech). In this research, POS information is required to identify aspects which occur as nouns. Third, lemmatisation and stemming is performed on the aspects. The aim of lemmatisation and stemming is to reduce inflectional (or derivationally related) forms of a word to a common base form. However, stemming employs crude heuristics to achieve its goal which in some instances can result in the token losing its meaning. In contrast, lemmatisation removes inflectional endings and returns the dictionary form of a word called lemma. Thus, the resultant aspects are meaningful and suitable to use for product representation. For example, given the word *buy*, stemming will return *bui*, whereas lemmatisation would maintain the word *buy*. In this research, lemmatisation is used to remove redundant aspects extracted from the dependency representation. Further, all text is set in lower case to avoid inaccuracies that occur due to mixed cases for the same content (e.g. "CAMERA" and "camera").

It is important to note that the dependency representation needs to be built before text pre-processing, as swapping these two steps will lead to imprecisions caused by incorrect POS assignments. Figure 5.2 compares sample outputs for the example sentence "*It doesn't matter if the lens sold with it isn't weather sealed*" before and after text pre-processing. The example on the left is not pre-processed and the example on the right is lemmatised. In these examples, the dependency representation for each sentence is shown below the sample sentence and every word in the sentence is attached with its POS. For example, "It/PRP" means the POS for the word "It" is a personal pronoun (PRP).



Raw Text

"It/PRP doesn/VBZ 't/RB matter/VB if/IN the/DT lens/NN sold/VBN with/IN it/PRP is/VBZ n't/RB weather/VB sealed/VBN ./."

The Stanford Dependencies representation:

nsubj (matter-4 , It-1)
nsubj (is-11 , lens-7)
acl (lens-7 , sold-8)
nmod (sold-8 , it-10)
advcl (matter-4 , is-11)

Lemmatised

"It/PRP do/VBP not/RB matter/VB if/IN the/DT lens/NN sell/NN with/IN it/PRP be/VB not/RB weather/VB seal/NN ./."

The Stanford Dependencies representation:

nsubj (matter-4 , It-1)
compound (sell-8 , lens-7)
nsubj (weather-13 , sell-8)
nmod (sell-8 , it-10)
advcl (matter-4 , weather-13)
dobj (weather-13 , seal-14)

Figure 5.2: Comparison of Dependency Representation between Raw Text and Lemmatised Text

Here, the word *sold* has *sell* as its lemma. However, the POS for *sold* is a verb and *sell* is a noun. Such differences impact the dependency rules that get triggered which in turn impact the extraction of aspects. In particular, in the lemmatised sentence, the relation *compound* is found because both *lens* and *sell* appear together as a noun (*compound* is a relation that relates two nouns that appear together). Hence, the parser will make mistakes by considering both words as compound nouns. Therefore, it is crucial to feed the raw text as input to the Stanford CoreNLP for processing.

5.2 Aspect Extraction Using Selected Dependency Relations

The computational implementation of dependency relations for English is the Stanford CoreNLP. The dependency representation output by the Stanford CoreNLP[1] is based on the Universal Dependencies (UD)[2] representation that was built using an English corpus. The English corpus consists of 254,830 words and 16,622 sentences which are taken from weblogs, newsgroups, emails, reviews and Yahoo! answers. Each dependency representation consists of a dependency relation and the two words that are related by the dependency relation. Stanford CoreNLP has a total of 47 dependency relations. However, not all relations are useful for aspect extraction for product reviews.

A review of the literature in dependency relation models for aspect extraction in Chapter 2 shows that it is important to consider sentiment knowledge in the selection of relevant dependency relations. In this thesis, a method to perform informed selection of dependency relations using sentiment knowledge is proposed. A primary difference between the proposed approach and the work of other state-of-the-art relation-based approaches is that instead of using a small subset of dependency relations, the proposed approach considers all dependency relations available in the Stanford CoreNLP parser, making the number of dependency relations used to extract aspects higher than previous works. To validate the relevance of a dependency relation, a list of possible POS patterns that are generated by the dependency relation from an English corpus[2] are obtained. To do this, $patternPOS(dp)$ is used to retrieve the set of possible POS patterns for a given dependency relation dp where:

$$patternPOS(dp) = \{(pos_{x_1}, pos_{y_1})_1, (pos_{x_2}, pos_{y_2})_2, ..., (pos_{x_n}, pos_{y_n})_n\} \tag{5.1}$$

Here, pos_x and pos_y are the part-of-speech (POS) of the two words related by dp and $dp \in DP$. Based on the list of possible patterns, the following function was formalised

[1] http://nlp.stanford.edu/software/dependencies_manual.pdf

[2] http://universaldependencies.org/en/dep/index.html

to retain relations that are relevant in extracting product aspects:

$$RelevantDP = \{dp \in DP : isRelevant(dp)\} \tag{5.2}$$

$$isRelevant(dp) = \exists t \in Patterns : t \in patternPOS(dp) \tag{5.3}$$

$$Patterns = \{(N, N), (N, V), (V, N), (N, J), (J, N), (N, RB), (RB, N)\} \tag{5.4}$$

where $RelevantDP$ is a set of dp that satisfy the condition $isRelevant$. The condition $isRelevant$ is true if there exists a pattern t in a set of predefined patterns, $Patterns$, such as $patternPOS(dp)$ is true. Here, $Patterns$ is a set of POS patterns that is used to identify relevant dependency relations. In product reviews, there is a relationship between an aspect and the sentiment expressed on the aspect (Liu, 2015). In this work, aspects are considered to be nouns and sentiment words to be adjectives, verbs and adverbs, a convention which has been widely adopted in previous work (Hu and Liu, 2004, Popescu and Etzioni, 2007). Therefore, dependency relations that frequently relate nouns (N and N), noun and adjective (N and J), noun and verb (N and V) and noun and adverb (N and RB) are relevant for aspect extraction. For example, *picture quality* is related by the relation ***compound*** (Noun Compounds) and both terms are noun. Therefore, ***compound*** has a pattern of (N, N). Based on this condition, *isRelevant* filters out 9 out of 47 dependency relations leaving 38 dependencies for evaluation. The list of selected dependency relations is summarised in Table 5.1.

Patterns	**Dependency Relations**
(N,J) (J,N)	acl, acl:relcl, advcl, amod, appos, advmod, ccomp, compound, conj, csubj, dep, det, discourse, dislocated, dobj, goeswith, list, mark, name, nmod:npmod, nmod:tmod, nsubj, nsubjpass, nummod, parataxis, remnant, vocative, xcomp
(N,V) (V,N)	cop, csubjpass, cc, case, iobj, reparandum
(N,RB) (RB,N)	cc:preconj, expl, neg
(N,N)	compound

Table 5.1: Selected Dependency Relations

The Stanford CoreNLP parser takes in a review sentence and produces a list of word pairs for each dependency relation. Let S be the set of sets of word pairs output from the Stanford CoreNLP parser where $S = \{s_1, s_2, ..., s_n\}$. The representation for each s is a set of pairs as follows:

$$s = \{(w_{x_1}, w_{y_1})_1, (w_{x_2}, w_{y_2})_2, ..., (w_{x_n}, w_{y_n})_n\} \tag{5.5}$$

where each pair is composed of the words w_x and w_y from the sentence which are related by a selected dependency relation $sdp \in RelevantDP$. Noun terms that are related by selected dependency relations are considered as potential aspects. This condition is implemented by $DirectRelations$ as follows:

$$DirectRelations_{sdp}(S) = \{a_1, a_2, ..., a_n\} \tag{5.6}$$

Accordingly, the final output of $DirectRelations$ is a set of aspects, a, extracted from the selected dependency relations.

5.2.1 Rule-based Frequent Noun Approach

The approach proposed in previous section extracted more than 5,000 unique aspects for every dataset. The number of extracted aspects is extremely high because potential aspects extracted from dependency representation are not all genuine aspects. For instance, the noun *salesperson* in the sample sentence - "*The salesperson is easy going and gave me a good discount*" will be extracted in the proposed approach because the noun *salesperson* depends on the adjective *easy* through the dependency relation ***nsubj***. This shows that additional heuristic rules are required to identify meaningful aspects and filter spurious aspects. Such heuristics are described in the rest of this section.

Pruning of technical specifications. In electronic product reviews, users may express their opinions by describing technical details of an aspect. For example, *Sigma 18-250mm lens* is describing the size of the camera lens. During parsing, the size of the lens (18-250mm) is automatically parsed as a noun but the size range of the lens is not an aspect. Technical specifications which are erroneously extracted as potential aspects are removed. Specifically, special characters and numbers tagged as nouns are removed.

Global frequency pruning. Information presented in product reviews tends to be highly dynamic and written in an informal manner leading to sparse and infrequent nouns. Misspelled words and web addresses often get parsed as nouns and are typical examples of this problem. To overcome this limitation, frequency-based pruning is conducted to retain aspects whose frequency is above a specific threshold in the product reviews. Here, the frequency threshold is set to 2 which is commonly adopted in the literature (Hu and Liu, 2004, Qiu et al., 2011). Furthermore, by applying this rule, the noun '*salesperson*' in the previous sample sentence will likely to be filtered since the sentence is describing the experience of a user not related to the product aspects, therefore it is unlikely to be a frequent word.

Co-occurrence of Aspect and Sentiment Words in the Same Sentence. Not all nouns extracted from product reviews are aspects. One solution proposed by Hu and Liu (2004) is to remove aspects that are not associated with sentiment words which exist in a manually crafted sentiment lexicon[1]. In this work, this approach is adapted to remove aspects that do not co-occur within the same sentence with sentiment words. Here, the sentiment words are identified by using SmartSA.

5.3 Generating Recommendation

Aspect extraction provides the context for recommendation, as such aspects can be seen as a means to represents products which allow product comparisons at the aspect level. A product can be represented using the vector space model (VSM) in a n-dimensional space, where each dimension corresponds to a separate aspect. If an aspect occurs in the reviews of a product, its value in the vector is a non-zero value. Accordingly a product, p is represented as a vector of aspects:

$$p = [a_k, a_{k+1}, a_{k+2}...a_n] \tag{5.7}$$

Here, a_k is the value for an aspect and n is the size of the vector. In order to compare products on the basis of aspects and the general sentiment about these aspects; each a_k, instantiates the general sentiment expressed in reviews about a_k. This provides a product representation that is based on aspect sentiment scores.

Figure 5.3 details the aspect sentiment scoring process. As discussed in Chapter 2, the most appropriate approach in identifying the sentiment word of an aspect is through word distance. Given the extracted aspects, the sentiment word with the minimum distance (minimum number of words) from the aspect is the target sentiment word of the aspect (step 4). Once the target sentiment word is identified, the next step is to determine the polarity of the sentiment. The aspect sentiment scoring process begins with pre-processing the review text using the standard text pre-processing steps. This includes tokenization, POS tagging and lemmatisation. In aspect extraction, it is important to ensure all aspects are in lower case to avoid the aspect extraction algorithm recognising the same content written in different formats (e.g. "CAMERA" and "camera"). However, in sentiment analysis the tokens are not converted into consistent case to preserve the sentiment expressed through capitalisation. For instance, capitalisation of a word such as "GREAT" will have different degree of sentiment intensity from its lowercase equivalent "great". In addition, stop words filtering is not necessary since stop-words are typically not included in a lexicon or are zero valued in high-coverage lexicons (e.g. SentiWordNet) and thus will not influence polarity classification.

The sentiment score of an aspect is generated using a state-of-the-art lexicon-based sentiment analysis system, SmartSA (Muhammad et al., 2016). To do this, a window of words centred on the target sentiment word is extracted. It is presented to the tool for sentiment scoring in order to capitalise on the contextual analysis offered by SmartSA (step 5). Here, the text window size is set to 4 as recommended in the literature (Dong et al., 2016). The resulting scores from SmartSA determine the final orientation of the sentiment on each aspect at the sentence and product level. At the sentence level, the sentiment score, *SentiScore*, of an aspect is computed by the difference between a positive (Pos) and negative (Neg) score of the target sentiment word given by SmartSA (step 6). A high positive value of *SentiScore* means that the strength of the positive sentiment expressed is high; similarly a high negative value signifies the strength of the negative sentiment is high.

$$SentiScore = Pos - Neg \tag{5.8}$$

At the product level, sentiment scores for each unique aspect are aggregated using an arithmetic mean (step 9). Therefore, the sentiment score of an aspect at the product level is computed as follows:

$$ProductScore(p_i, a_j) = \frac{\sum_{j=1}^{|\mathcal{A}^i|} AspectSentiScore(p_i, a_j)}{|\mathcal{A}^i|} \tag{5.9}$$

$$AspectSentiScore(p_i, a_j) = \frac{\displaystyle\sum_{m=1}^{|\mathcal{R}_j^i|} SentiScore(r_m)}{|\mathcal{R}_j^i|} \tag{5.10}$$

Where R_j^i is a set of reviews for product p_i related to aspect a_j and $r_m \in R_j^i$. Here, $AspectSentiScore$ allows the sentiment of product, p_i, to be associated with individual aspects $a_j \in \mathcal{A}^i$. $\mathcal{A}^i$ is the subset of aspects shared between the query and candidate product.

5.4 Evaluation of Dependency-based Aspect Extraction

The aim of this evaluation is to conduct a comparative study of the proposed aspect extraction methods against the baseline methods discussed in Chapter 4 to ascertain the effectiveness of the proposed approach in a product recommendation setting. The comparative study includes the following baseline approaches:

- FREQ: a combination of shallow NLP and frequent noun approach to extract aspects (Dong et al., 2016). When implementing this strategy, the threshold for frequency pruning is set to 30% as implemented in previous work (Dong et al., 2016).

```
Input:     $\mathcal{A}^i$, Set of aspects shared between the query and candidate product
           $\mathcal{R}$, Set of reviews
Output:    $AspectSentiScore$, Sentiment scores for an aspect $a$
Required:  SmartSA
for each $a \in \mathcal{A}^i$ do
    Initialise $SentimentWords$
    for each $sentence \in \mathcal{R}$ do
        Find the nearest sentiment word to $a$
        $SentimentWords \leftarrow$ applyWindowText{sentiment word}
        Compute $SentiScore$ from SmartSA using $SentimentWords$
        $Sum+ = SentiScore$
    end for
    $AspectSentiScore = Sum/|\mathcal{R}^i_j|$
end for
return $AspectSentiScore$
```

Figure 5.3: Aspect Sentiment Scoring

- MogDP: a set of nine dependency relation rules is used to extract aspects. Thereafter, the set of candidate aspect phrases is pruned using a frequency cut-off (Chen et al., 2014). The list of relations are *amod, acomp, nsubj, cop, dobj, compound, conj* and *neg* (see Section 2.1.5).

- SenticNetDR: capitalises on common-sense knowledge and a set of manually defined dependency relation rules to extract aspects (Poria et al., 2014). The list of relations includes *advmod, amod, advcl, xcomp, cop, cc, conj, dobj, prep* and *compound* (see Appendix B).

- DirectRelations: implements the proposed approach without the rule-based frequent noun approach (Equation 5.6).

- DirectRelations$^+$: implements the improved method in Equation 5.6, which combines DirectRelations with the rule-based frequent noun approach described in Section 5.2.1.

5.4.1 Results of Dependency Rule-based Aspect Extraction Approach

Tables 5.2 and 5.3 show the MAP and RI results achieved in *k*-fold cross validation over seven datasets. Numbers in bold indicate the best performance in a dataset. The results of a Friedman test show that there is a significant difference between the algorithms (at $\alpha = 0.05$) for all datasets in MAP and RI with the p-value of 0.004 and 0.008 respectively. It can be seen from Table 5.2 that DirectRelations$^+$ has achieved the best performance across all datasets in MAP with a 17% improvement on average compared to other baselines. The post-hoc test results for MAP in Table 5.4 show that DirectRelations$^+$ performs significantly better than all the dependency-based methods (MogDP, SenticNetDR) in MAP.

In Table 5.3, it can be observed that DirectRelations$^+$ recommends products with a relative rank improvement between 7.0% and 27.7%. Given the retrieval set of 15 products, this means that DirectRelations$^+$ is recommending products that are, on average, up to 4 rank positions better than the query product in terms of overall user ratings. In contrast, the RI for baseline approaches, in most cases, is less than 6.7%. Since recommending a product with one rank position better than the query product will result in 6.7% rank improvement, a RI lower than 6.7% suggests that the baseline approaches recommend products that rank below the test query product in most cases. Specifically, MogDP performs the worst among all the baselines as it has a RI of less than 6.7% in 5 out of 7 datasets. Post-hoc tests in RI (Table 5.5) show that DirectRelations$^+$ performs significantly better than MogDP and the p-value between SenticNetDR and DirectRelations$^+$ is 0.053 which approaches but does not reach conventional statistical significance (< 0.05). Although there is no significant difference observed between DirectRelations$^+$ and SenticNetDR, the RI results show that DirectRelations$^+$ is the best performing approach compared to other dependency-based baseline approaches. The superior performance of DirectRelations$^+$ in both MAP and RI demonstrates that combining the rule-based frequent noun approach with an informed selection of dependency relations improves recommendation performance.

Comparison between DirectRelations$^+$ and FREQ indicates that the former is superior to the latter in both MAP and RI in all the datasets. However, post-hoc results indicate that there is a significant difference between DirectRelations$^+$ and FREQ in MAP but not in RI. Based on the results in Table 5.3, it can be observed that DirectRelations$^+$

[3] The asterisk (*) in the table indicates there is a significant difference between the two approaches

Methods	DSLR	Laptops	Tablets	Phones	Printers	Mp3	Tv
FREQ	0.630	0.401	0.526	0.596	0.351	0.579	0.471
MogDP	0.657	0.401	0.526	0.514	0.344	0.411	0.506
SenticNetDR	0.638	0.404	0.493	0.584	0.312	0.617	0.485
DirectRelations	0.635	0.422	0.562	0.594	0.308	0.588	0.513
DirectRelations$^+$	**0.740**	**0.497**	**0.567**	**0.644**	**0.397**	**0.635**	**0.566**

Table 5.2: Results for MAP

Methods	DSLR	Laptops	Tablets	Phones	Printers	Mp3	Tv
FREQ	21.7	2.3	21.3	8.4	2.3	10.0	3.4
MogDP	18.9	5.5	20.6	3.0	3.4	2.5	2.3
SenticNetDR	15.1	4.0	20.6	8.3	3.9	10.4	2.7
DirectRelations	16.8	12.4	**24.0**	7.5	1.1	10.2	4.9
DirectRelations$^+$	**27.7**	**17.9**	22.5	**10.2**	**7.3**	**11.2**	**7.0**

Table 5.3: Results for RI(%)

Approaches	p-value
DirectRelations$^+$ - DirectRelations	0.18
DirectRelations$^+$ - FREQ	0.018*
DirectRelations$^+$ - MogDP	0.018*
DirectRelations$^+$ - SenticNetDR	0.013*
DirectRelations - FREQ	1.00
DirectRelations - MogDP	1.00
DirectRelations - SenticNetDR	1.00
FREQ - MogDP	1.00
FREQ - SenticNetDR	1.00
MogDP - SenticNetDR	1.00

Table 5.4: Post-hoc Test Results for MAP[3]

is able to consistently recommend products that are at least one rank position ($\geq$6.7%) better than the query product across all datasets. In contrast, FREQ is unable to recommend products that are '*better*' than the query product in Laptops, Printers and TV. Recall from Chapter 4 that a lower RI is more damaging as users are likely to feel disappointed with the recommender systems if they are recommended products that have lower ratings than the ones they are looking at. As such, although there is no significant

Approaches	p-value
DirectRelations$^+$ - DirectRelations	0.28
DirectRelations$^+$ - FREQ	0.112
DirectRelations$^+$ - MogDP	0.005*
DirectRelations$^+$ - SenticNetDR	0.053
DirectRelations - FREQ	1.00
DirectRelations - MogDP	1.00
DirectRelations - SenticNetDR	1.00
FREQ - MogDP	1.00
FREQ - SenticNetDR	1.00
MogDP - SenticNetDR	1.00

Table 5.5: Post-hoc Test Results for RI[3]

difference observed, DirectRelations$^+$ performs better than FREQ overall. A comparison between DirectRelations and DirectRelations$^+$ shows that the rule-based frequent noun approach plays a crucial role in improving recommendation performance when using a dependency relation approach. It can be observed in Table 5.2 and 5.3 that by combining the rule-based frequent noun approach to DirectRelations, recommendation performance is improved by at least 0.8% in MAP and 9% in RI. This suggests that combining the rule-based frequent noun approach with informed selection of dependency relations provides an improvement on recommendation performance over not combining them.

5.4.2 Analysis of Extracted Aspects

Results in Tables 5.2 and 5.3 show that FREQ, SenticNetDR and DirectRelations did not perform well compared to DirectRelations$^+$. In Figure 5.4, it can be observed that FREQ, SenticNetDR and DirectRelations extract a large number of aspects (ranging from 3,792 to 23,215). In contrast, the number of aspects extracted using DirectRelations$^+$ is fewer than 6,000 in a majority of the dataset except for Tablets which has 16,258 aspects extracted. One plausible reason for their poor performance is that these approaches extract a large number of aspects which increases the opportunity of using spurious aspects for product representation. Therefore, the poor performances of FREQ, SenticNetDR and DirectRelations suggest that recommendation performance does not benefit from approaches that produce high coverage of aspects.

Table 5.6 shows the top 10 most frequent aspects in DSLR. Aspect terms in bold and italic indicate genuine aspects. The genuine aspects were identified using two publicly available annotated camera review datasets[4]. Based on the list of aspects in Table 5.6, it can be observed that out of 10 aspects, the number of genuine aspects extracted by FREQ is 4, SenticNetDR and DirectRelations both extracted 7 genuine aspects, MogDP extracted 8 genuine aspects and DirectRelations$^+$ extracted 9 genuine aspects. It is clear that aspects extracted by FREQ, SenticNetDR and DirectRelations contain fewer genuine aspects than MogDP and DirectRelations$^+$. This is expected as extracting a large number of aspects increases the occurrence of spurious aspects. However, it is interesting to note that MogDP extracts more genuine aspects than FREQ, SenticNetDR and DirectRelations but performs poorly in RI (Table 5.3). Figure 5.4 shows that MogDP, which only applies 9 dependency rules, extracts the fewest aspects with an average of 1,012 aspects across all datasets. The poor performance of MogDP in RI suggests that additional dependency relations are required to extract aspects that improve recommendation performance.

Table 5.2 shows that the MAP score difference between DirectRelations and DirectRelations$^+$ is only 0.005 in Tablets and the RI results in Table 5.3 provide evidence that not applying the rule-based frequent noun approach yields a better recommendation performance for Tablets. One possible reason for this is that the rule-based frequent-noun approach does not recognise aspects that are semantically similar. For example, in camera reviews, reviewers may use *picture* or *photo* to refer to the image they took using the camera. Given that Tablets dataset has the highest number of reviews among all datasets (as shown in the dataset statistics in Table 4.1) and that different reviewers refer to the same aspect using different aspect terms (Dong et al., 2016), the varied terms that reviewers used to refer to the same aspect could have caused some of the genuine aspects to appear as infrequent nouns. Therefore, applying rule-based the frequent noun approach on Tablets dataset will remove some of the genuine aspects.

[4] https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

Methods	**Aspects**
FREQ	***camera***, ***lens***, canon, nikon, ***video***, time, ***quality***, ones, plot, point
MogDP	***camera***, ***lens***, ***picture***, ***quality***, ***feature***, ***shot***, ***photo***, ***price***, time, dslr
SenticNetDR	***camera***, ***purchase***, nikon, ***price***, ***lens***, ***picture***, ***work***, ***use***, love, lot
DirectRelations	***camera***, ***purchase***, ***lens***, ***price***, canon, time, ***picture***, hand, ***work***, ***use***
DirectRelations$^+$	***camera***, ***lens***, ***picture***, ***feature***, ***quality***, ***shot***, dslr, ***price***, ***light***, ***focus***

Table 5.6: Top 10 Most Frequent Aspects for DSLR (genuine aspects are italic and bold)

It can be observed that using a small number of dependency rules to extract aspects does not provide any improvement on recommendation performance. However, when using a high number of dependency rules to extract aspect, a large number of aspects will be extracted resulting in a higher chance of using spurious aspects for product representation. To overcome this limitation, the rule-based frequent noun approach is required to improve recommendation performance by filtering spurious aspects. However, the rule-based frequent noun approach does not perform well in datasets which have a large number of reviews. This is because reviewers tend to use different aspect terms to refer to the same aspect and the rule-based frequent noun approach is unable to detect these aspect terms as being interrelated. As a result, some of the genuine aspects are removed due to being infrequent and this will cause a drop in recommendation performance.

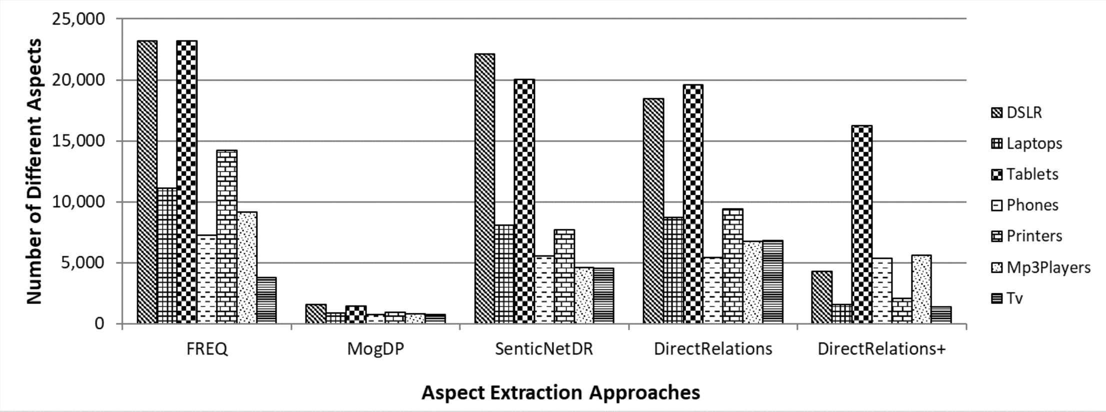


Figure 5.4: Number of Aspects Extracted by Different Aspect Extraction Methods

The number of aspects extracted using DirectRelations$^+$ ranges from a minimum 1,386 to a maximum of 16,258. Typically all extracted aspects from reviews are used in recommendation. However, it is not realistic to assume that all aspects are equally important to users when making a purchase decision. Therefore, the next chapter presents an aspect weighted sentiment scoring strategy and an analysis of the recommendation performance of the proposed approach.

5.5 Chapter Summary

This chapter presents an informed aspect extraction approach for social recommender systems which exploits sentiment knowledge and the frequent noun approach to select a subset of dependency rules to extract genuine aspects. In addition, a heuristic rule is proposed to prune spurious aspects such as technical specifications and web addresses that are extracted from the dependency relations.

The evaluation of the aspect extraction approaches shows that the proposed dependency rule-based aspect extraction approach performs best in MAP and RI compared to other dependency-based methods. This suggests that combining an informed selection of dependency relations with a rule-based frequent noun approach can effectively extracts aspects that are relevant in product representation. A comparison between frequency-based method (FREQ) and the proposed DirectRelations$^+$ shows that DirectRelations$^+$ performs consistently better than FREQ across all datasets. Although there is no significant difference between these two approaches in RI, the RI results indicate that DirectRelations$^+$ is able to recommend '*better*' products in all datasets whereas FREQ is only able to recommend '*better*' products in four out of seven datasets. This further emphasises the superiority of the proposed DirectRelations$^+$ approach. Analysing the aspects extracted by each aspect extraction approach suggests that recommendation performance does not benefit from approaches that produce a high coverage of aspects as this will increase the opportunity of having spurious aspects in the product representation. Therefore, the rule-based frequent noun approach is required. However, one limitation observed in DirectRelations$^+$ approach is that this approach does not perform well on large datasets as it is not able to recognise the different ways that reviewers used to refer to the same aspect. As a result, some of the genuine aspects are removed by the rule-based frequent noun approach and this will cause a drop in recommendation performance. This observation is only recognised in one of the datasets, therefore further experiments is required to ascertain this finding.

Chapter 6

Preference-based Aspect Weights

Key to accurate recommendation is to have the right product representation. Social content such as product reviews offers an opportunity to enhance recommender algorithms by exploiting user feedback. A survey of the literature suggests that review-based recommender systems tend to focus on using explicit feedback such as product reviews to generate recommendation. However, user interactions are a key indicator of their preferences which manifest as explicit and implicit user feedback. This chapter introduces a novel approach to product recommendation by harnessing social content from users' explicit and implicit feedback: product reviews and users' product purchase preferences. Specifically, the proposed approach infers important aspects from a preference graph built from users' purchase preferences and sentiment extracted from product reviews. To this end, an aspect-weighted sentiment scoring algorithm is formalised to score the products and rank them for recommendation purposes.

6.1 Users feedback

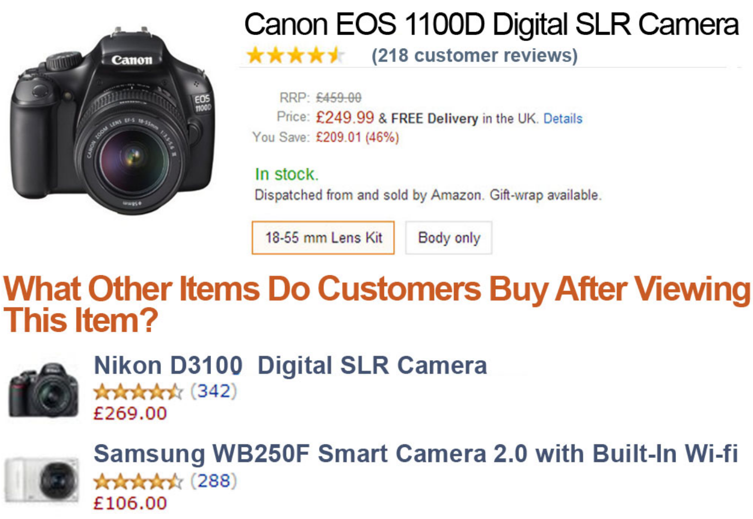
Product reviews are an explicit form of feedback whilst purchase preferences are an implicit form of feedback. This section discusses the characteristics of each type of feedback and justifies the purpose of combining both sources of knowledge as a way to score a product.

6.1.1 Explicit User Feedback

Knowledge embedded in user-generated content has been emphasised as valuable resources for recommendation systems (Dong et al., 2016, Wang and Chen, 2012). In Chapter 5, an approach that utilises the sentiment of product aspects expressed in reviews for product representation was presented. However, consumers often use different terms to refer to the same aspect in product reviews. Therefore, a product may have hundreds of aspects and each with a different level of importance to different consumers (Zha et al., 2014). This becomes a challenge when recommending products to new users (e.g. cold-start users) when their preferences are not known by the system. Therefore, additional sources of information are needed to help identify important aspects that influence users' purchase decisions.

6.1.2 Implicit User Feedback

Preference knowledge can potentially be used to improve recommendations as illustrated in Figure 6.1. Here in addition to typical information about *Canon EOS 1100D Digital SLR Camera* (e.g. Camera image and textual description), there is also information about user preferences (e.g. what users typically buy after viewing this camera). It can be observed that *Nikon D3100 Digital SLR Camera* (*NikonSLR*) and *Samsung WB250F Smart Camera* (*SamsungSmart*) are products that many users purchased after viewing *CanonSLR*. Based on this information, two preference relations are generated in which *NikonSLR* is preferred over *CanonSLR* and *SamsungSmart* is preferred over *CanonSLR*. Such list of purchased products provides valuable insights about the preference of users. Therefore, the preference relations on products are an indicator of aspect importance. This is because purchase choices are based on comparison of products, which involves a comparison of the aspects of these products. In particular a user's purchase preferences hints at aspects that are likely to have influenced their purchase decision and as such be deemed more important. Therefore, the proposed approach captures all preference relations between products using a preference graph and analyses this structure to infer aspect importance.



Canon EOS 1100D Digital SLR Camera

(218 customer reviews)

RRP: ~~£459.00~~

Price: £249.99 & FREE Delivery in the UK. Details

You Save: £209.01 (46%)

In stock.

Dispatched from and sold by Amazon. Gift-wrap available.

18-55 mm Lens Kit

Body only

What Other Items Do Customers Buy After Viewing This Item?

Nikon D3100 Digital SLR Camera

(342)

£269.00

Samsung WB250F Smart Camera 2.0 with Built-In Wi-fi

(288)

£106.00

FIGURE 6.1: Product information.

6.1.3 Preference Graph

A preference relation between a pair of products denotes the preference of one product over the other through the analysis of viewed and purchased product relationship. Figure 6.2 illustrates a preference graph, $G = (\mathcal{P}, \mathcal{E})$, generated from a sample of Amazon data on DSLR camera. The number of reviews/questions for a product is shown below each product node. The set of nodes, $p_i \in \mathcal{P}$, represent products, and the set of directed edges, $\mathcal{E}$, are preference relations. A directed edge from product p_i to p_j with $i \neq j$ represents that, for some users, p_j is preferred over product p_i and is represented as $p_j \succ p_i$. In some cases where $p_j \succ p_i$ and $p_i \succ p_j$, a bidirectional preference relation can be observed. For any p_i, $\mathcal{E}^i$ denotes incoming and $\mathcal{E}_i$ for outgoing product sets.

Typically products that have many incoming edges are more popular, while less popular products tend to have more outgoing edges. Therefore, it is expected that product Nikon D3100 in Figure 6.2 is listed in Amazon's Best Seller Ranked list in Figure 6.3. Here, N/A means that the products were not ranked. It can be observed from Figure 6.3 that products ranked at the top have higher number of incoming links than the rest of the products. For instance, in Laptops, the product that is ranked as the top 1 product has the highest number of incoming links. Similar observations can be made for Printer and TV where products with the highest number of incoming links are ranked at the top. However while the assumption is true with most studied products, it is not always the

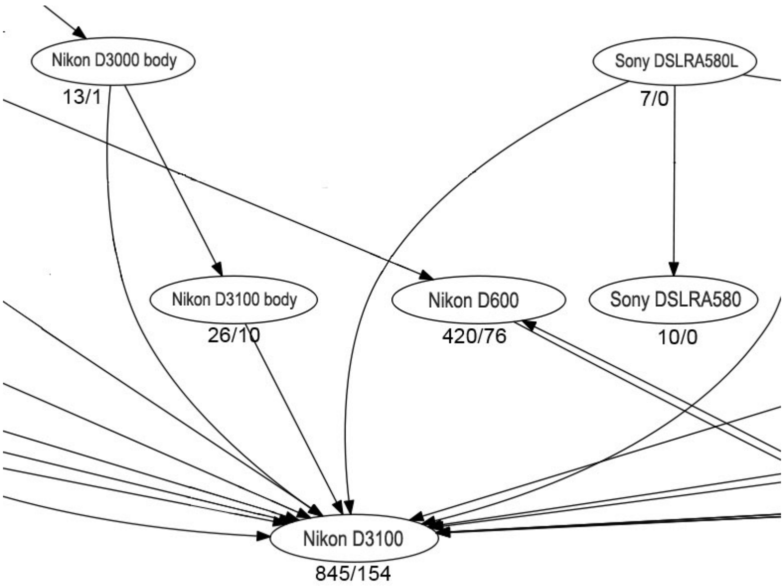


FIGURE 6.2: Preference sub-graph for *Amazon Digital SLR Cameras.*

case that a product with higher number of incoming links will always have a higher rank in Amazon's Best Seller ranked list. For example, with Mp3Players, Phones and Tablets, products which are in the lower rank (e.g. rank 4 to 5 in Tablets) has a higher number of incoming links than the higher ranked product. This shows that popularity of a product is not the only reason that consumers purchase products. Therefore, this motivates the need to leverage further dimensions of knowledge sources, such as sentiment, from online reviews for product recommendation.

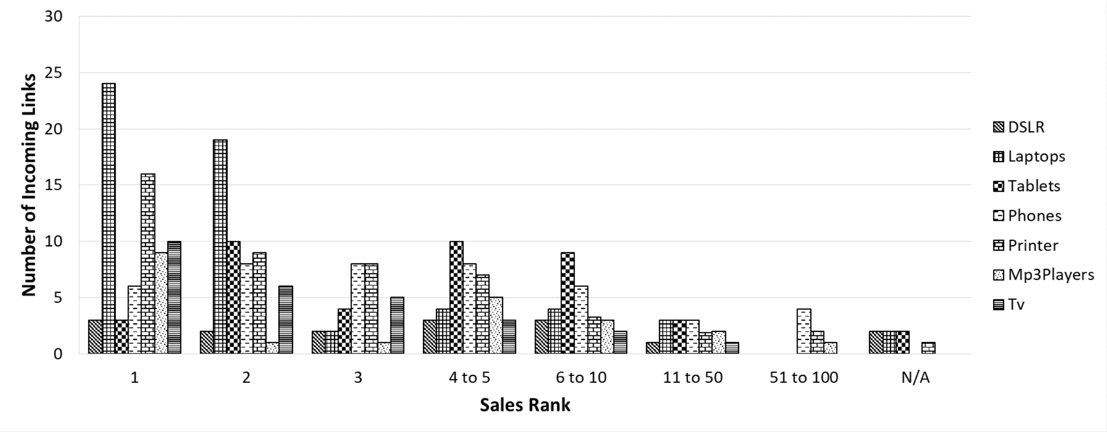


FIGURE 6.3: Amazon Best Seller Rank

6.2 Aspect Weighted Sentiment Scoring

Reviews are authored following the purchase of products and contain user opinion in the form of positive and negative sentiment. The strength of sentiment expresses the intensity with which an opinion is stated with reference to a product (Turney, 2002). This information is exploited as a means to rank the products, such that products ranked higher are associated to a higher positive sentiment for aspects deemed important. Therefore, a finer-grained analysis of reviews by computing sentiment at the aspect level is performed. Given a query product Q, the *ProductScore* of a candidate product, p_i, given a set of related reviews is computed as a weighted summation of sentiments expressed at the aspect level as follows:

$$ProductScore(p_i, a_j) = \frac{\sum_{j=1}^{|\mathcal{A}^i|} AspectWeight(a_j) * AspectSentiScore(p_i, a_j)}{\sum_{j=1}^{|\mathcal{A}^i|} AspectWeight(a_j)} \tag{6.1}$$

where *AspectSentiScore* is the sentiment scores of the aspect, a_j, derived from product reviews and *AspectWeight* is the aspect importance weights learned by comparing the sentiment difference between node pairs in the preference graph. Here, $\mathcal{A}^i$ is a subset of aspects that are shared between the query and candidate product.

6.2.1 Preference-based Aspect Weight Extraction

Aspects frequently associated with a positive sentiment in purchased products but with a neutral or negative sentiment in viewed products will naturally be considered as important aspects with respect to the users' purchase decisions. Based on this principle, in Equation 6.2 *AspectSentiScore′* of aspect a in product p is 1 if *AspectSentiScore* is greater than a threshold h and 0 otherwise, where the default value for h is 0.

$$AspectSentiScore'(p, a) = \begin{cases} 1, & \text{if } AspectSentiScore(p, a) > h; \\ 0, & \text{otherwise.} \end{cases} \tag{6.2}$$

Learning aspect weights relies on the preference graph and aspect level sentiment knowledge. A product purchase choice is a preference made on the basis of one or more aspects. The notion of aspect importance arises when the same set of aspects contribute to similar purchase decisions. Using this same principle, aspect weights are derived by comparing the aspect sentiment score differences between viewed and purchased product pairs. For each product pair, the preference difference between any pair of products is computed as:

$$\delta'(a_j, p_x, p_y) = AspectSentiScore'(p_x, a_j) - AspectSentiScore'(p_y, a_j) \tag{6.3}$$

$$\delta(a_j, p_x, p_y) = |L_{min}(\mathcal{A}, \mathcal{E})| + \delta'(a_j, p_x, p_y) \tag{6.4}$$

The preference difference score is given in Equation 6.3 and the score is adjusted using Equation 6.4 to avoid having negative aspect weights. Here $|L_{min}(\mathcal{A}, \mathcal{E})|$ is the lowest preference difference score obtained over all the aspects for all product preference pairs. Finally, the aspect weights are computed by aggregating aspect sentiment differences between viewed and purchased product pairs in the training set as follows:

$$AspectWeight(a_j) = \frac{\sum_{x=1}^{|\mathcal{P}|} \sum_{y=1}^{|\mathcal{P}|} \delta(a_j, p_x, p_y)}{|t \in \mathcal{E}|} \tag{6.5}$$

Here $(p_x, p_y) \in \{(p_x, p_y)\}_{x \neq y}^{t}$ where either $p_x \succ p_y$ or $p_y \succ p_x$ or both, $\mathcal{P}$ is the set of products in the training set, t is the set of product preference pairs containing aspect a_j and $\mathcal{E}$ is the set of all product preference pairs.

The proposed preference-based aspect weighting approach is illustrated with the following example. Figure 6.4 and 6.5 illustrates the notion of preference difference calculations using a trivial three node preference graph. In Figure 6.4, the relation $p_3(+0.5) \succ p_1(-0.8)$ denotes that product p_3 is preferred over p_1 and they have an aspect sentiment score of $+0.5$ and -0.8 respectively for aspect *lens*. A similar explanation holds for the aspect *screen* in Figure 6.5. Corresponding preference difference scores are shown in Table 6.1 for the two aspects. Here p_3 has a sentiment score greater than 0 and p_1 has a sentiment score of less than 0. Based on the condition in Equation 6.2, $AspectSentiScore'$ for $p_3 = 1$ and $p_1 = 0$. The preference difference score is computed using Equation 6.3. In Table 6.1, it can be observed that $p_6 \succ p_5$ for aspect *screen* has a score of -1. Therefore, Equation 6.4 is applied to avoid having negative aspect weights. Next, the sentiment difference between the product pairs is aggregated using Equation 6.5. As a result, *lens* and *screen* have a normalised aspect weights of 2.0 and 0.5 respectively. This suggests that aspect *lens* is more important than aspect *screen*.

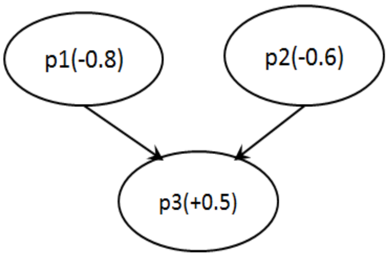


FIGURE 6.4: Sub-graph *lens* aspect.

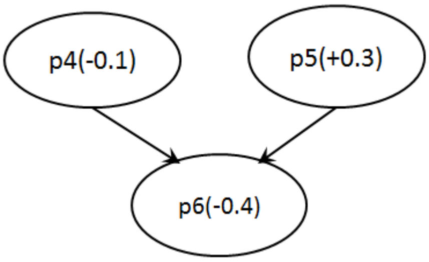


FIGURE 6.5: Sub-graph *screen* aspect.

Aspects (a_j)	**Preference Relations**	$\delta'(a_j, p_x, p_y)$	$\delta(a_j, p_x, p_y)$	$AspectWeight(a_j)$
lens	$p_3 \succ p_1$	1 - 0 = 1	1 + 1 = 2	$\frac{4.0}{2} = 2.0$
	$p_3 \succ p_2$	1 - 0 = 1	1 + 1 = 2	
screen	$p_6 \succ p_4$	0 - 0 = 0	1 + 0 = 1	$\frac{1.0}{2} = 0.5$
	$p_6 \succ p_5$	0 - 1 = -1	1 + (-1) = 0	

TABLE 6.1: Aspect preference scores.

6.2.2 Sentiment Distribution

The overall opinion of a product's reviews can be expressed as a distribution over each aspect. Measuring the distribution of sentiments of an aspect is helpful in determining the positivity or negativity of an aspect. Here, two statistical measures designed to quantify inequality in arbitrary distributions are proposed.

6.2.2.1 Gini Coefficient

Opinions formed by consumers are not always the same. When some consumers may appreciate an aspect of a product, others may criticise it. To quantify the opinion of an aspect, the most common technique is to compute the arithmetic mean of the aspect sentiment scores (Wang and Chen, 2012, Wang and Wang, 2014). However, in some cases reviewers who voted strongly in favour of an aspect may overpower those of others with a less strong opinion. Consider the distribution of sentiments for Camera A's aspects in Figure 6.6. Aspect *image* has 1 positive sentiment and 4 negative sentiments. Based on this information, it is reasonable to say that Camera A has a bad *image*. However, after aggregating all the sentiments of aspect *image* given by the reviewers, the result is a positive average sentiment. A closer look at each sentiment given by the reviewers in Figure 6.7 shows that reviewer R3 expressed a strong positive sentiment ($SentiScore =$ 0.28) whereas other reviewers express weak negative sentiments ($SentiScore < 0.1$). This shows that the arithmetic mean is not a useful aggregator of opinions from a pool of reviewers.

Another common approach to aggregate opinions is to count the number of positive and negative sentiments (Zhang et al., 2012) for each aspect. However, this approach does not consider the strength of the sentiment when assigning sentiment polarity of an aspect. For example, an aspect with a sentiment score of 0.005 is a very weak positive and most likely will not give much information and therefore should be categorised as neutral. Therefore, the question that is addressed here is how to detect if there is social agreement about the sentiment expressed about an aspect.

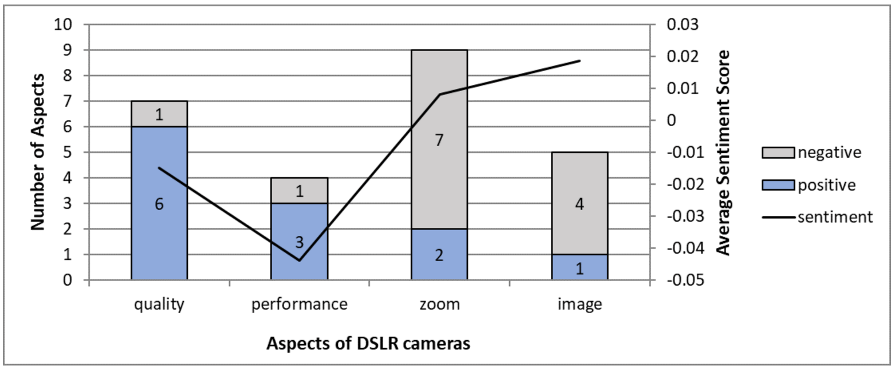


FIGURE 6.6: Aspect-Sentiment Distribution for Camera A

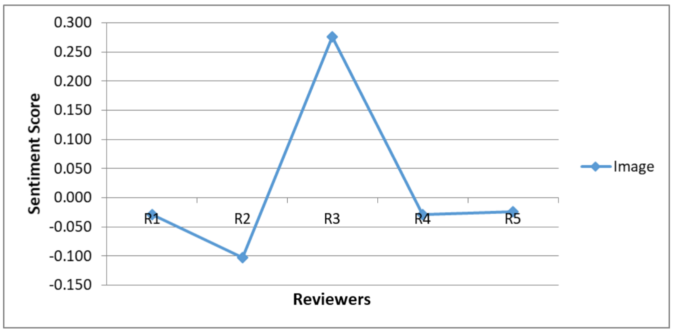


FIGURE 6.7: Sentiment Score for aspect *image*

The Gini coefficient is a measure of inequality in a population which is commonly used in the analysis of wealth distribution. In this work, Gini scores are used to measure the social agreement of sentiment from users who commented on an aspect of a product. Formally, given a list of sentiment scores for an aspect ranked in ascending order, the Gini coefficient is defined as follows (Dixon et al., 1987):

$$Gini = \frac{\sum_{i=1}^{f}(2i - f - 1)x_i}{f\sum_{i=1}^{f} x_i} \tag{6.6}$$

where f is the size of the list of sentiment scores for the aspect and x is the observed sentiment score for the aspect which has rank i. A Gini score of 0 represents a total equality where all sentiment scores of an aspect are the same, which also means that there is a social agreement on the sentiment expressed. In contrast, a Gini score of 1 represents maximal inequality where there is no social agreement on the sentiment expressed (e.g. all observed sentiment scores of an aspect have a value of 0 except one which has a non-zero value.)

6.2.2.2 Wilson Interval

The absolute number of times an aspect is being mentioned in a product's reviews matters. For example, an aspect observed to have 2 positive sentiments and 1 negative sentiment may have the same positive average sentiment score as an aspect which has 5 positive sentiments and 4 negative sentiments. However, the latter carries more significance because it was measured over a larger sample of data points. This means that the confidence in this aggregated score being representative of what the whole population of reviewers would think is higher. To take into account the statistical significance of the average sentiment score, the following Wilson interval (Wilson, 1927) is proposed:

$$WI = \frac{1}{1 + \frac{1}{n}z^2}[\hat{p} + \frac{1}{2n}z^2 \pm z\sqrt{\frac{1}{n}\hat{p}(1-\hat{p}) + \frac{1}{4n^2}z^2}] \tag{6.7}$$

$$WI = (v - \sigma, v + \sigma) \tag{6.8}$$

In Equation 6.7, z is the $1-\frac{1}{2}\alpha$ quantile of the standard normal distribution, α is the error percentile (here $\alpha = 5\%$ and the confidence level is set to 95%), n is the sample size and $\hat{p}$ is the observed proportion. The rationale behind using WI is to measure statistical significance of the given average sentiment score. Therefore, the value $\hat{p}$ depends on the polarity of the average sentiment score of an aspect, where $\hat{p} = \frac{s}{n}$. For instance, if the average sentiment score of an aspect is positive, then s is the number of positive sentiments observed in the reviews for the aspect. A similar approach is applied when the observed score is negative.

Instead of taking the arithmetic average of the sentiment score, WI gives a range in which the real score might lie with a 95% confidence. The output of WI is a lower bound and a upper bound value as shown in Equation 6.8 where $v - \sigma$ is the lower bound value and $v + \sigma$ is the upper bound value. In order to measure the statistical significance, it is important to take into account both bounds and the size of the interval between them. This is because although a larger v indicates significance of the sentiment score, a larger interval indicates the sample size is small and as such is less significant. Therefore, a smaller interval (σ) with a larger v represents a more significant average sentiment score. The approach proposed by Zhang et al. (2015) to measure the significance of the average sentiment score assigned to an aspects by using the mid-point of the lower bound ($v - \sigma$) and the interval centre v is thus adopted in the rest of this work. The $WilsonScore$ for the average sentiment of an aspect is calculated as follows:

$$WilsonScore = v - \frac{1}{2}\sigma \tag{6.9}$$

The difference between *Gini* and *WilsonScore* is that *Gini* measures the inequality of the sentiment distribution without taking into account the frequency of the aspect term in reviews. In contrast, the *WilsonScore* takes into account the frequency of the aspect to measure the confidence of the average sentiment score. In this research, both metrics are used as sources of evidence to assign higher sentiment scores to an aspect when there is consensus about the distribution of the sentiment and otherwise is penalised accordingly. The *AspectSentiScore* in Equation 5.10 is modified to combine either *Gini* or *WilsonScore* weighting as follows:

$$AspectSentiScore(p_i, a_j) = \frac{\sum_{m=1}^{|\mathcal{R}_j^i|} SentiScore(r_m)}{|\mathcal{R}_j^i|} * (1 - Gini) \tag{6.10}$$

$$AspectSentiScore(p_i, a_j) = \frac{\sum_{m=1}^{|\mathcal{R}_j^i|} SentiScore(r_m)}{|\mathcal{R}_j^i|} * (WilsonScore) \tag{6.11}$$

6.3 Preliminary Observation: Product Pairs Similarity

The aspect extraction algorithm extracted more than a thousand unique aspects for each product domain (e.g. DSLR, Laptops, Tablets, Phones, Printers, Mp3Players and TV). It is expected that the aspects extracted are sufficient to generate rich product cases as there are on average 119 - 1,010 aspects to describe a product. However, it is important to assess if there is a sufficient number of shared aspects between the query and candidate product since a low number of shared aspects between products would result in sparse representations and poor comparisons.

Figures 6.8 to 6.14 show the distribution of the similarity values between all product pairs for each product domain. It can be observed that there is a wide range of similarity values which means that products can be compared with different characteristics (aspects). Specifically, the mean similarity values for Tablets, Printers, Mp3Players and TV are high (between 0.41 and 0.53) whilst DSLR, Laptops and Phones are low (less than 0.4). A low similarity value means that there is a very small number of shared aspects between product pairs. To ensure there is a sufficient number of aspects for comparison, the percentage of products that shared k number of aspects for DSLR, Laptops and Phones in Figure 6.15 is further examined. Based on these results, it can be observed that 87-96% of the product pairs from DSLR and Phones share more than 20 aspects. In contrast, Laptops have a much narrower distribution: 70% of product pairs share more than 20 aspects and 20% of the product pairs share between 10 and 20 aspects. This indicates that users expressed their opinion on a smaller number of aspects in Laptops. From these observations, it can be concluded that majority of the product pairs share at least 10 aspects. The fact that there are at least 10 shared aspects is reassuring since it means that sensible comparisons can be made.

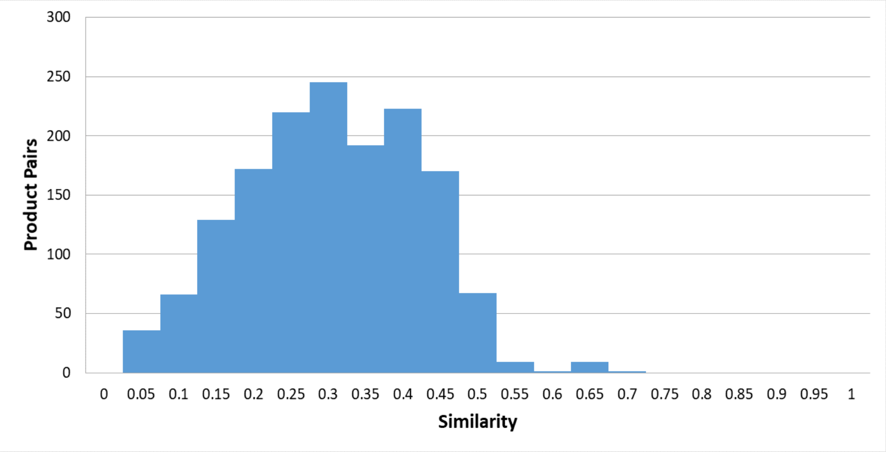


FIGURE 6.8: DSLR ($\mu = 0.38, \sigma = 0.12$)

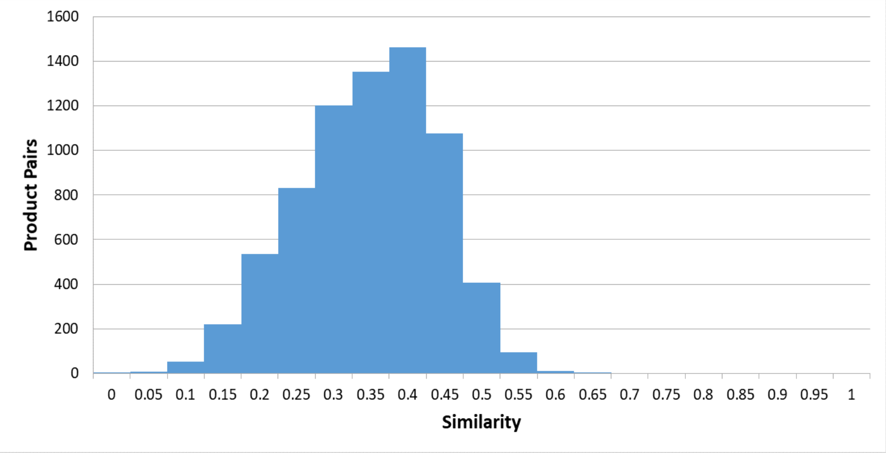


FIGURE 6.9: Laptops ($\mu = 0.37, \sigma = 0.09$)

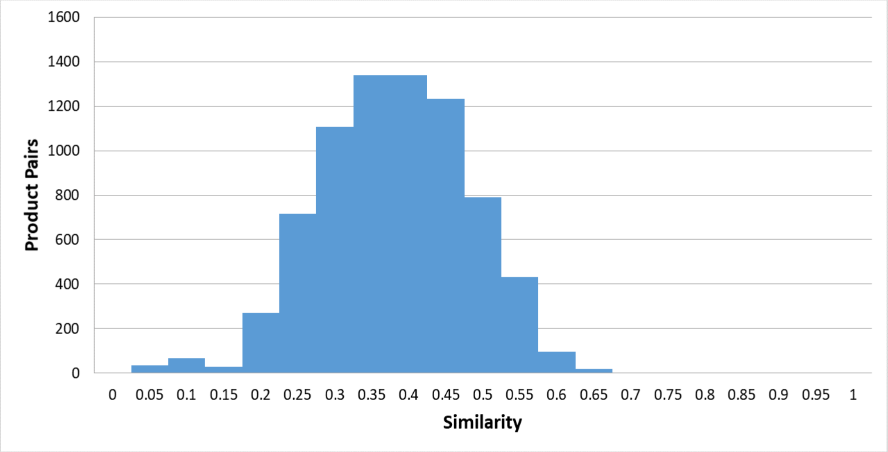


FIGURE 6.10: Tablets ($\mu = 0.41, \sigma = 0.10$)

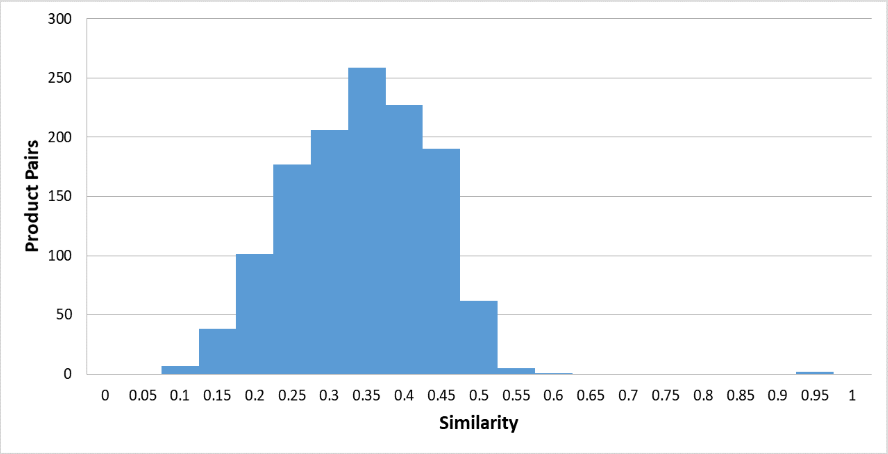


FIGURE 6.11: Phones ($\mu = 0.37, \sigma = 0.09$)

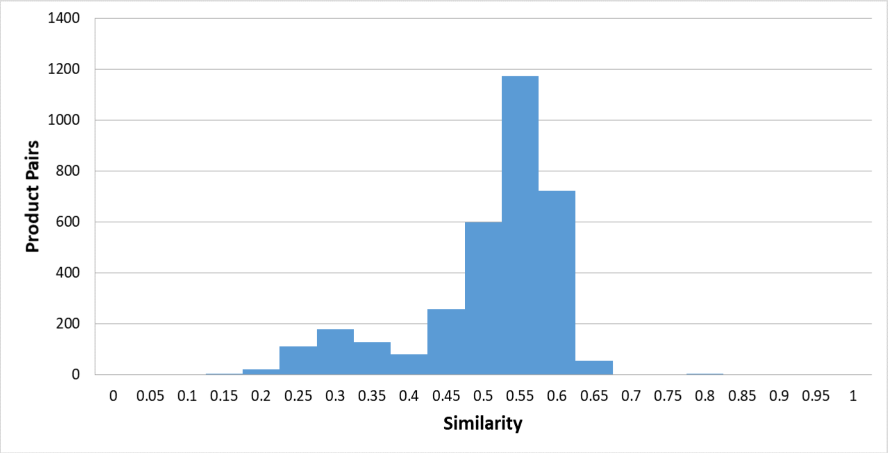


FIGURE 6.12: Printers ($\mu = 0.53, \sigma = 0.10$)

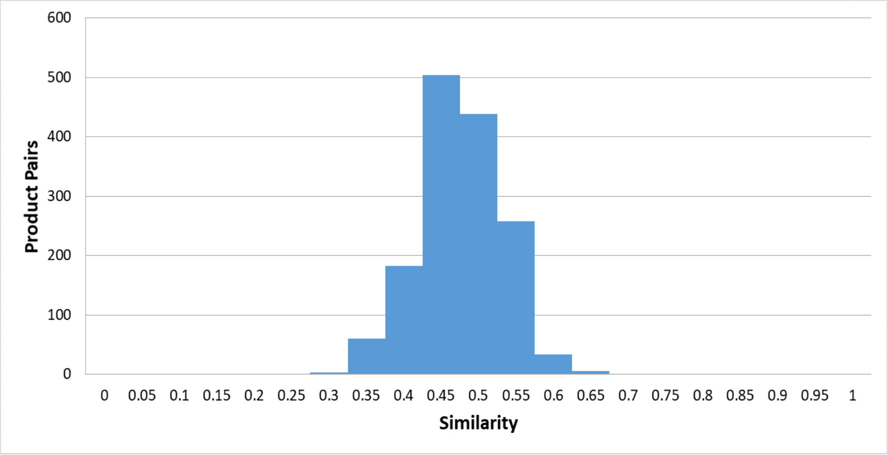


FIGURE 6.13: Mp3Players ($\mu = 0.50, \sigma = 0.06$)

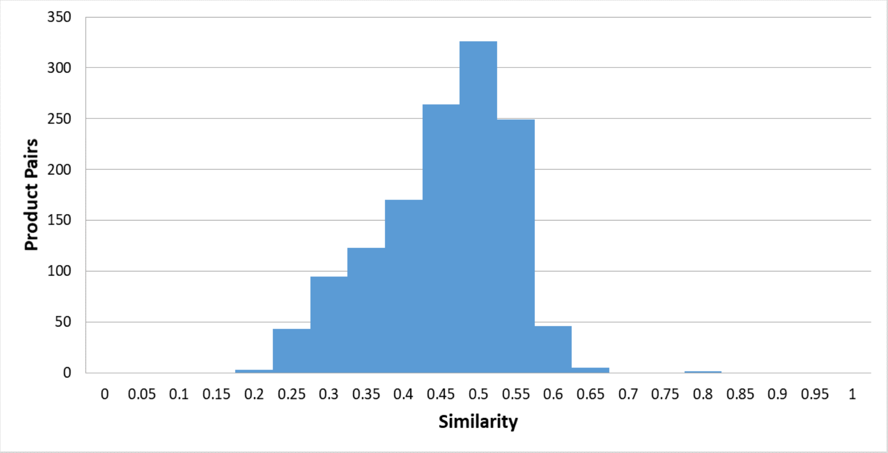


FIGURE 6.14: TV ($\mu = 0.48, \sigma = 0.09$)

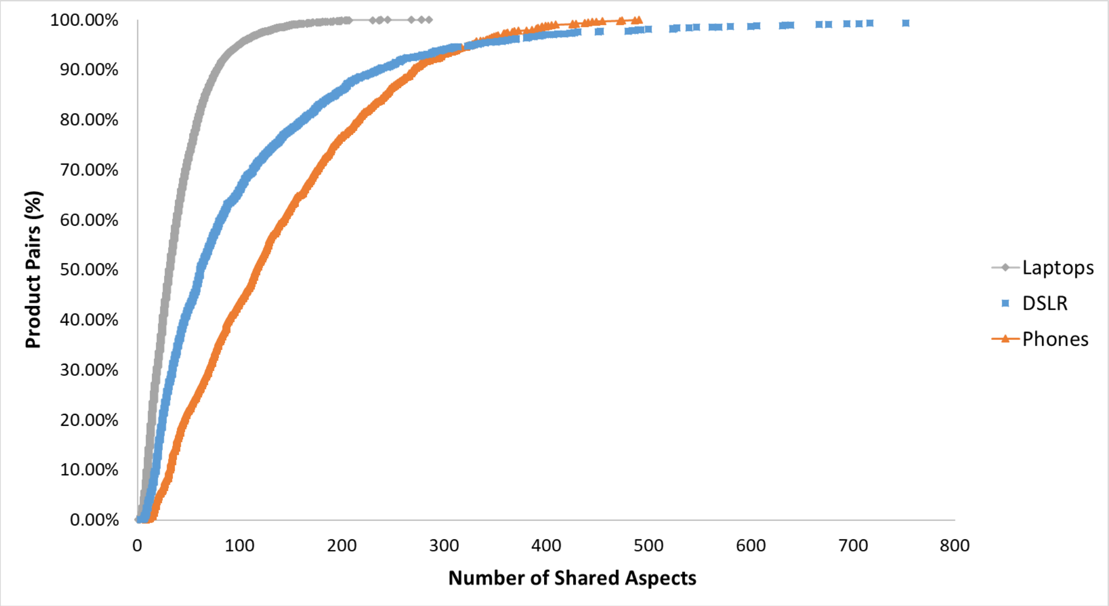


Figure 6.15: Distribution of Shared Aspects per Product Pair

Methods	**DSLR**	**Laptops**	**Tablets**	**Phones**	**Printers**	**Mp3**	**TV**
Cosine	0.398	0.357	0.389	0.393	0.379	0.443	0.445
BetterScore	0.674	0.484	0.492	0.600	0.353	0.548	0.550
PageRank	0.295	0.375	0.403	0.366	0.365	0.618	0.394
Pref	0.720	0.498	0.555	0.634	0.360	0.613	0.584
Pref+ST	**0.734**	0.517	**0.567**	0.651	**0.44**	**0.621**	**0.61**
Pref+Gini	0.727	0.500	0.545	0.633	0.371	0.617	0.577
Pref+Wilson	0.704	**0.541**	0.543	**0.652**	0.348	0.580	0.571

Table 6.2: MAP with Preference based Aspect Weights

6.4 Evaluation on Preference-based Aspect Weights

The aim of this evaluation is to study the performance of the proposed aspect weighted sentiment scoring strategies introduced in this chapter. The first experiment aims to ascertain the performance of the proposed aspect weighted sentiment scoring strategy in comparison to existing state-of-the-art methods. The second experiment determines whether combining preference-based aspect weights with an analysis of sentiment distribution improves recommendation performance. To this end, the following variations of the proposed aspect-weighted sentiment scoring strategies to compute $ProductScore$ are used:

- Pref: Uses preference-based aspect weights and aspect sentiment scores when generating $ProductScore$ without considering sentiment threshold.
- Pref+ST: Uses Pref with sentiment threshold (Equation 6.2).
- Pref+Gini: Combines Pref with Gini coefficient (Equation 6.10).
- Pref+Wilson: Combines Pref with Wilson score (Equation 6.11).

[1] The asterisk (*) in the table indicates there is a significant difference between the two approaches

Methods	**DSLR**	**Laptops**	**Tablets**	**Phones**	**Printers**	**Mp3**	**TV**
Cosine	12.72	-0.21	6.83	0.71	1.93	-1.57	-2.65
BetterScore	21.61	13.83	17.57	7.68	9.33	8.70	6.71
PageRank	6.23	3.45	14.89	-9.04	4.21	**16.57**	8.25
Pref	25.80	15.42	20.97	10.20	8.54	11.90	9.19
Pref+ST	**26.67**	17.43	**22.51**	**10.4**	**10.09**	11.3	**12.61**
Pref+Gini	24.62	15.11	21.80	10.10	8.77	11.06	9.06
Pref+Wilson	25.01	**19.12**	21.58	10.35	7.01	9.26	11.58

Table 6.3: RI(%) with Preference based Aspect Weights

6.4.1 Results of Aspect Weighted Sentiment Scoring Approaches (Pref and Pref+ST)

Tables 6.2 and 6.3 list the results in terms of MAP and RI respectively on the seven datasets. As before, bold font indicates the best performance on a dataset. The three baselines used for comparisons are the following:

- Cosine is a similarity-based approach using cosine similarity metric to rank products (see Section 4.2.1).
- BetterScore is a state-of-the-art approach that utilises users' sentiments in a content-based recommender system. (see Section 4.2.2).
- PageRank is a popularity-based approach that recommends products based on the popularity of products in a graph (see Section 4.2.3).

Results from Tables 6.2 and 6.3 show that Pref+ST achieved the best results in DSLR, Tablets, Printers and TV in both MAP and RI. In Laptops, best results are observed with Pref+Wilson with a MAP score of 0.541 and RI of 19.12%. In Phones, the best MAP score is achieved by Pref+Wilson but in RI, the best performing approach is Pref+ST. However, it can be observed in Table 6.2 that the MAP score difference between Pref+ST and Pref+Wilson is only 0.001. Therefore, it can be said that both approaches achieved similar results in MAP for Phones. Although Pref+ST performed well in majority of the datasets, Pref+ST did not perform as expected in Mp3. In particular, the RI results

Approaches	p-value
Pref+ST - Pref+Gini	1.000
Pref+ST - Pref	1.000
Pref+ST - Pref+Wilson	0.744
Pref+ST - BetterScore	0.011*
Pref+ST - PageRank	0.004*
Pref+ST - Cosine	0.002*
Pref+Gini - Pref	1.000
Pref+Gini - Pref+Wilson	1.000
Pref+Gini - BetterScore	1.000
Pref+Gini - PageRank	0.545
Pref+Gini - Cosine	0.280
Pref - Pref+Wilson	1.000
Pref - BetterScore	1.000
Pref - PageRank	1.000
Pref - Cosine	0.545
Pref+Wilson - BetterScore	1.000
Pref+Wilson - PageRank	1.000
Pref+Wilson - Cosine	1.000
BetterScore - PageRank	1.000
BetterScore - Cosine	1.000
Cosine - PageRank	1.000

Table 6.4: Post-hoc Test Results for MAP[1]

in Table 6.3 for Pref+ST were unexpectedly poor in the Mp3 dataset compared to just using PageRank, and the MAP score in Table 6.2 for Pref+ST (0.621) is close to the PageRank score (0.618). This requires further analysis on the extracted aspects in Mp3 dataset to identify the possible cause of its poor performance. This analysis will be discussed in Section 6.4.3.

The results of a Friedman test show that there is a significant difference between the algorithms with under 5% significance level assumptions for all datasets with a p-value close to 0 ($p < 0.0001$) in both MAP and RI. The post-hoc test results for MAP and RI in Tables 6.4 and 6.5 show that there is a significant difference between Pref+ST and Cosine as well as Pref+ST and PageRank. MAP and RI results in Tables 6.2 and 6.3 show that the aspect weighted sentiment driven approach with sentiment threshold

Approaches	p-value
Pref+ST - Pref+Gini	1.000
Pref+ST - Pref	1.000
Pref+ST - Pref+Wilson	1.000
Pref+ST - BetterScore	0.063
Pref+ST - PageRank	0.011*
Pref+ST - Cosine	0.000*
Pref+Gini - Pref	1.000
Pref+Gini - Pref+Wilson	1.000
Pref+Gini - BetterScore	1.000
Pref+Gini - PageRank	1.000
Pref+Gini - Cosine	0.136
Pref - Pref+Wilson	1.000
Pref - BetterScore	1.000
Pref - PageRank	0.744
Pref - Cosine	0.027*
Pref+Wilson - BetterScore	1.000
Pref+Wilson - PageRank	0.744
Pref+Wilson - Cosine	0.027*
BetterScore - PageRank	1.000
BetterScore - Cosine	1.000
Cosine - PageRank	1.000

TABLE 6.5: Post-hoc Test Results for RI[1]

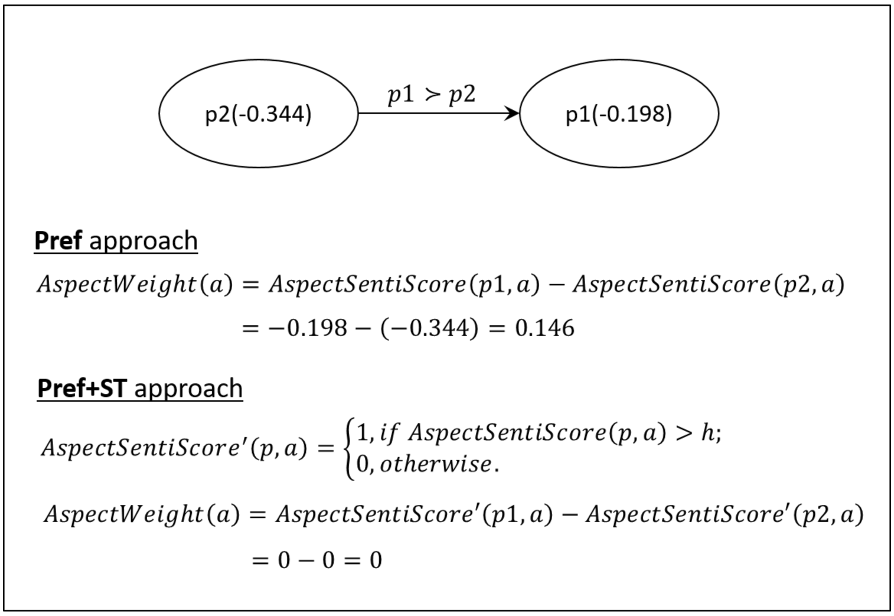
(Pref+ST) performs better than Cosine and PageRank which does not consider the sentiment of aspects, instead ranking products by measuring similarity of aspects. This finding supports those reported in Dong et al. (2016) where similarity-based approaches that do not consider sentiment of aspects fail to recommend products that are higher ranked than the query product. Post-hoc tests results for MAP and RI show that there is a significant difference between Pref+ST and BetterScore in MAP but not in RI. Although there is no significant difference observed in RI, comparing sentiment values of aspects between products did not contribute to *'better'* recommendation performance compared to Pref+ST. This is demonstrated in Table 6.3 where Pref+ST consistently achieved higher RI score than BetterScore across all datasets with an average recommendation performance improvement of 34%. Among all datasets, the highest improvement observed (88%) is in TV where Pref+ST and BetterScore achieved a RI of 12.61% and 6.71% respectively. This shows that BetterScore which does not consider the importance of aspects had a disadvantage in ranking products. As a result, Pref+ST outperform BetterScore in both evaluation metrics.

Aspect weights with preference knowledge and sentiment threshold (Pref+ST) provide the most improvement on a majority of the datasets (5 out of 7). Specifically, the improvement gain from Pref+ST over Pref is up to 22.2% and 37.2% for MAP and RI respectively. This shows that setting sentiment threshold for each product preference pair to determine it's polarity is superior to the approach that does not consider sentiment threshold in estimating aspect weights. This suggests that the importance of an aspect can be ascertained by observing the differences between its sentiment values in purchased-viewed product pairs.

Furthermore, there is another benefit of setting a sentiment threshold in aspect weighting. Consider an example given in Figure 6.16, where an aspect a only appears in the reviews of products p_1 and p_2, where p_1 is preferred over p_2. The sentiment scores of a for products p_1 and p_2 are -0.198 and -0.344 respectively. It is reasonable to say that a is not a good aspect in both products and there is no evidence to suggest that a is an important aspect. Therefore, a weight of 0 should be assigned to a. In contrast, the preference difference score between p_1 and p_2 for aspect a is a positive difference (0.146) in the Pref approach. Therefore, setting a sentiment threshold helps overcome the issue caused by negative sentiment scores.

6.4.2 Results of Aspect Weighted Sentiment Scoring with Sentiment Distribution Analysis (Pref+Gini and Pref+Wilson)

This subsection investigates whether measuring the distribution of sentiments of an aspect has an impact on recommendation performance. Table 6.6 shows the performance difference when comparing Pref with Pref+Gini and Pref+Wilson. It can be observed that Pref+Gini achieved 0.3% improvement on average in MAP when comparing to Pref. However, there are no improvement observed in RI in majority of the datasets as the average performance difference between Pref and Pref+Gini is -1.3%. Since RI is valued



Pref approach

$$AspectWeight(a) = AspectSentiScore(p1, a) - AspectSentiScore(p2, a)$$
$$= -0.198 - (-0.344) = 0.146$$

Pref+ST approach

$$AspectSentiScore'(p, a) = \begin{cases} 1, if\ AspectSentiScore(p, a) > h; \\ 0, otherwise. \end{cases}$$

$$AspectWeight(a) = AspectSentiScore'(p1, a) - AspectSentiScore'(p2, a)$$
$$= 0 - 0 = 0$$

Figure 6.16: Comparison between Pref and Pref+ST

Method	Metric	DSLR	Laptops	Tablets	Phones	Printers	Mp3	TV	Average
Pref vs. Pref+Gini	MAP	1.0%	0.4%	-1.8%	-0.2%	3.1%	0.7%	1.2%	0.3%
	RI	-4.6%	-2.0%	4.0%	-1.0%	2.7%	-7.1%	-1.4%	-1.3%
Pref vs. Pref+Wilson	MAP	-2.2%	8.6%	-2.2%	2.8%	-3.3%	-5.4%	-2.2%	-0.6%
	RI	-3.1%	24.0%	2.9%	1.5%	-17.9%	-22.2%	26.0%	1.6%

Table 6.6: Performance Difference Comparison of Pref vs. Pref+Gini and Pref vs. Pref+Wilson

over MAP, the results in Table 6.6 suggests that Pref+Gini does not improve Pref. This might be explained by analysing the Gini score for each aspect. In Figure 6.17, the Gini scores for aspects are concentrated between 0.5 and 0.6. Since a Gini score of 1 indicates no social agreement on the sentiment, a Gini score greater than 0.5 indicates little social agreement on the sentiment expressed on a majority of the product aspects. Therefore, Gini has little effect on the recommendation performance.

Results for Pref+Wilson is also mixed. Overall, Table 6.6 shows that Pref+Wilson does not improve Pref in MAP in majority of the datasets but there is a 1.6% improvement on average in RI. Specifically, it can be observed that Pref+Wilson yields the highest improvement on Laptops in MAP (8.6%) and RI (24.0%) and performs the worst in

Mp3 in both MAP (-5.4%) and RI (-22.2%). Recall from Section 6.2.2.2 that the Wilson score takes into account the size of the Wilson interval. A large interval indicates that the frequency of occurrence of an aspect in product reviews is low. In contrast, a small interval indicates that the frequency is high in product reviews. Therefore, the marginal performance of Pref+Wilson might be explained by the frequency of aspects in the dataset. Figure 6.18 shows the Wilson interval sizes for aspects in Laptops and Mp3. Here, Mp3 has a very high percentage of aspects that have low frequency compared to Laptops. Specifically, 66.8% of aspects in Mp3 have an interval size greater than 0.8 compared to 32.4% of aspects in Laptops. The limited occurrence of each unique aspect in the Mp3 dataset limits the opportunity of the Wilson score to improve recommendation performance.

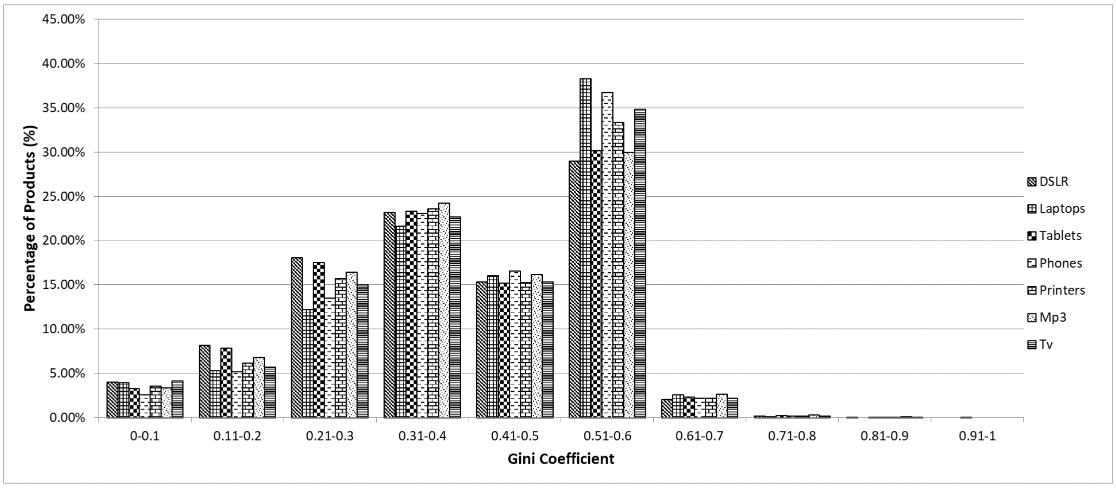


Figure 6.17: Gini Coefficient for All Aspects

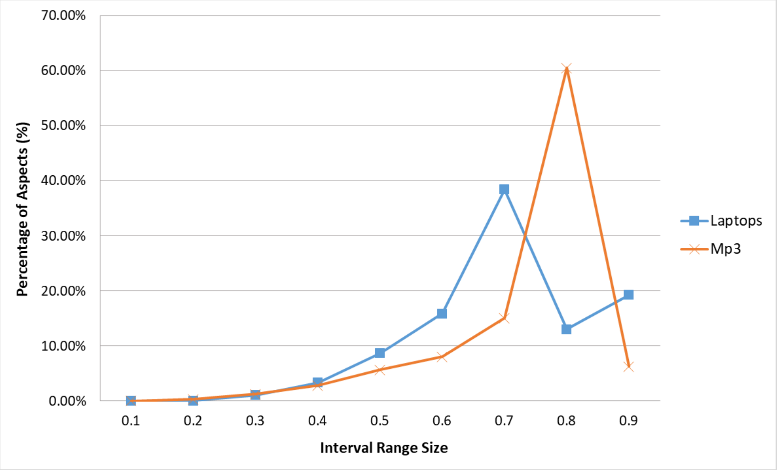


Figure 6.18: Wilson Interval Size

6.4.3 Further Analysis on Extracted Aspects

The MAP and RI results in Tables 6.2 and 6.3 show that Pref+ST were unexpectedly poor in Mp3. This poor performance might be explained by the number of shared aspects between query and candidate products. In the Mp3 dataset, the number of shared aspects is consistently higher than that with other datasets. Specifically, the average minimum number of shared aspects for DSLR, Laptops, Tablets, Phones, Printers and TV datasets is between 1 and 21 ($\mu = 8.8$, $\sigma = 7.1$). However, the minimum number of shared aspects in Mp3 is 80, which is the highest among all datasets. In Section 5.4.2, the analysis of extracted aspects suggests that the recommendation performance does not benefit from a large number of aspects because this increases the opportunity of having spurious aspects in product representation. To ensure the aspects extracted are relevant in product representation, the list of aspects that are frequently shared between query and candidate products were examined. Table 6.7 shows the list of the most frequently shared aspects between query and candidate products in all product categories. Here, it can be observed that there are a number of spurious aspects. For instance, the terms *take*, *like* and *thing* are not aspects of a product. The term *thing* is a noun but it is not an aspect. Similarly, *like* (in lemma form) is not an aspect but it was extracted because the original word in the reviews is *likes*, which is a noun. Further, the parser does not perform well on sentences with grammatical errors. For instance, the sentence "*the loading take hours in my laptop*" causes the parser to tag *take* as a noun. This suggests that a higher number of shared aspects increases the opportunity of using spurious aspects when computing the $ProductScore$. This motivates the comparative study in the next section, which studies the impact of aspect selection using feature selection methods.

6.5 Chapter Summary

This chapter introduced a novel preference-guided aspect sentiment scoring algorithm for product recommendation. The novelty of this algorithm is that it integrates two knowledge sources: user product reviews and users' product purchase preferences. The

DSLR	*Laptops*	*Tablets*	*Phones*	*Printers*	*Mp3Players*	*TV*
camera	laptop	use	use	printer	product	picture
lens	buy	tablet	phone	ink	battery	this tv
this camera	screen	buy	buy	product	thing	quality
buy	work	work	time	set	make	buy
picture	like	time	work	print	quality	sound
take	price	price	like	quality	mp3	screen
like	time	screen	battery	buy	work	set
photo	this laptop	well	look	work	music	work
quality	keyboard	product	purchase	this printer	play	remote
feature	window	one	quality	set up	charge	price

Table 6.7: Top 10 Most Frequent Shared Aspects

combination of these knowledge sources was formalised by integrating the user preference on aspects and sentiment of these aspects. First, this chapter discussed the construction of the preference graph from users' purchase preferences. Thereafter, a preference-based aspect weighting algorithm that infers aspect weights from the preference graph was introduced. Third, the formalisation of the preference-guided aspect sentiment scoring algorithm to rank the products was described. And finally, the enhancement of the algorithm with a support factor using Gini coefficient and Wilson score was proposed.

Experiments were conducted to establish contributions of aspect weights, Gini coefficient and Wilson interval. Thereafter, the performance of the proposed approaches were compared against a similarity-based ranking, PageRank algorithm and the state-of-the-art approach that utilises users' sentiments in a content-based recommender system. Evaluation results show that combining users' product purchase preferences and sentiment knowledge can effectively improve recommendation performance in both MAP and RI. In particular, better performance is observed when setting a sentiment threshold. Furthermore, results on applying the Gini score in the recommendation algorithm show no improvement when comparing with the aspect weighted sentiment-driven approach (Pref). An analysis of the Gini score in the dataset shows that there is little social agreement on the sentiment expressed on a majority of the product aspects. Thus, Gini has little effect on recommendation performance. In contrast, evaluation results for Wilson score are mixed. Further analysis on the Wilson score shows that the limited occurrence of unique aspects limits the opportunity of the Wilson score to improve recommendation performance.

Chapter 7

Aspect Selection

User-generated content contains many non-standard lexical items and syntactic patterns. When reasoning with text in such circumstances one cannot rely on state-of-the-art NLP systems which are best placed to operate on formal language that adheres to static grammar rules. Therefore, the use of standard NLP tools for aspect extraction is likely to have a negative impact on precision. In Chapter 5, the proposed dependency rule based aspect extraction algorithm was built to recognise product aspects using syntactic patterns. Evaluation results show that the large number of aspects being extracted is detrimental to follow-on recommendations. In this situation identifying useful aspects is not straightforward and calls for heuristics that can help select or filter aspects from the initial aspect extraction step.

Survey of related work suggests that aspect-based recommender systems often tend to ignore the importance of aspect selection and for those that do consider this challenge use only frequency-based heuristics. This chapter studies the need for aspect selection more closely in the context of product recommendation and considers what knowledge sources might be used to develop selection heuristics (i.e. beyond frequency counting). For this purpose, feature selection methods commonly adopted by text classification systems, where feature selection helps dimensionality reduction and improves class discrimination, are used. With recommender systems, dimensionality reduction is crucial in identifying relevant aspects for product representation whilst discrimination will help separate relevant from less relevant aspects. A new recommendation strategy that integrates an approach to select important aspects in product representation is presented. In particular, this approach capitalises on feature selection techniques to infer aspect importance and thereafter rank the aspects for selection. Further, a weighted strategy that combines aspect weighting and aspect selection to rank products is presented.

7.1 Motivation for Aspect Selection

The aspect extraction algorithm described in Chapter 5 extracted more than 1,300 unique aspects for every product domain. In Table 7.1, it can be observed that there are on average more than 100 aspects extracted per product in each category.

Descriptions	**DSLR**	**Laptops**	**Tablets**	**Phones**	**Printers**	**Mp3**	**TV**
No. of products	56	121	122	51	82	55	52
Rating variance	0.06	0.16	0.24	0.11	0.09	0.51	0.08
No. of unique aspects	4298	1553	19,566	5379	2077	6771	1386
Average no. of aspects per product	394	119	1010	444	276	668	206

Table 7.1: Descriptive Statistics of Datasets

Typically all extracted aspects from reviews are used in recommendation. However, it is not realistic to assume that all aspects are of equal importance for a purchase decision. This is because there are ordinary nouns that are not aspects but are extracted as such due to parsing errors. Table 7.2 shows the list of top 10 most frequent aspects gathered from the evaluation datasets. Here, 2 to 5 most frequent aspects are not relevant. For instance, the aspect *time* appears in all product domains but *time* is not an aspect of a product. This shows that the aspect extraction algorithm can produce erroneous aspects, which will not benefit recommendation performance.

The negative effect of erroneous aspects on recommendation performance observed in the evaluation datasets is presented here. In the DSLR dataset, there are two camera products, P1 and P2, where P1 is ranked higher than P2 in the benchmark ranking. Table 7.3 shows the sentiment scores of each aspect for product P1 and P2. Here, aspect *camera* and *quality* are general aspects which users used to build their overall opinion

Top 10 Most Frequent Aspects						
DSLR	*Laptops*	*Tablets*	*Phones*	*Printers*	*Mp3Players*	*TV*
camera	laptop	tablet	phone	printer	player	tv
picture	window	screen	screen	set up	music	picture
lens	screen	buy	camera	time	review	sound
video	price	work	price	printing	product	price
thing	keyboard	app	app	ink	device	screen
time	time	price	battery	quality	quality	quality
feature	bit	battery	time	price	song	thing
shot	thing	product	quality	paper	price	color
quality	problem	easy	thing	problem	time	time
dslr	quality	time	bit	page	exchange	set up

Table 7.2: Top 10 Most Frequent Aspects

on the product. In contrast, *lens* and *photo* are specific aspects of a product. Further, *feature* and *dslr* were also erroneously extracted as aspects. To rank the products, *ProductScore* (without using aspect weights) was computed for each product using different combinations of aspects in Table 7.4. Here, there are six combinations of aspects[1]. Each combination was assigned an identification from A to F and used to compute *ProductScore* for P1 and P2. For instance, aspect combination B uses sentiment scores from aspects *camera* and *lens* to compute *ProductScore*.

ID	**Aspects**	**P1**	**P2**
1	camera	0.22	0.26
2	lens	0.60	0.30
3	quality	0.22	0.34
4	photo	0.15	0.18
5	feature	-0.31	-0.10
6	dslr	0.13	0.26

Table 7.3: Sentiment Scores of Aspects for Product P1 and P2

ID	**Aspects**
A	camera
B	camera, lens
C	camera, lens, quality
D	camera, lens, quality, photo
E	camera, lens, quality, photo, feature
F	camera, lens, quality, photo, feature, dslr

Table 7.4: Combination of Aspects

The results of using the combinations of aspects to compute *ProductScore* for P1 and P2 are presented in Figure 7.1. Based on the *ProductScore*, it can be observed that combinations B, C and D rank P1 higher than P2 whilst combinations A, E and F rank P2 higher than P1. Specifically, by adding sentiment scores of *feature* and *dslr*

[1] The number of aspects is determined based on the first occurrence of negative effects on recommendation performance

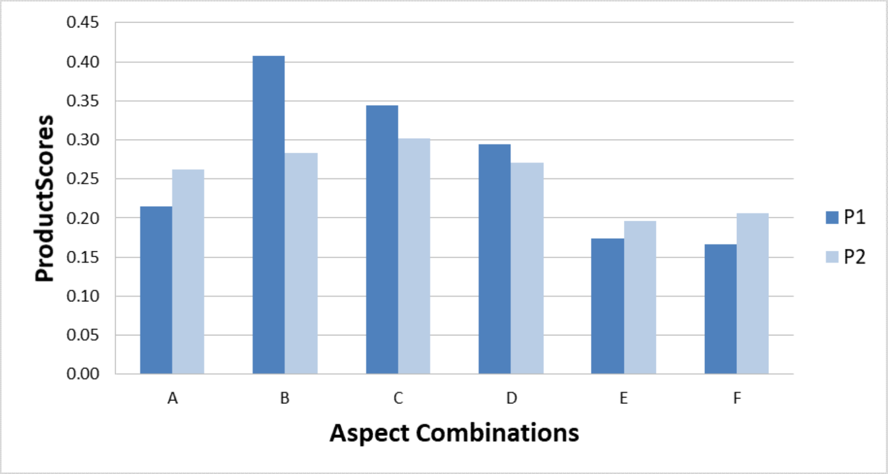


FIGURE 7.1: *ProductScore*

(combination E and F), P2 is ranked higher than P1. This demonstrates how adding irrelevant aspects can lead to poorer performance. Therefore, this calls for methods to select relevant aspects for recommendation, a form of feature selection.

The feature selection problem has been studied by the machine learning community to enhance accuracy in supervised learning tasks such as text classification. Feature selection is the process of selecting a subset of relevant features for model building. Its main aim is to remove redundant and irrelevant features that do not contribute to the predictive accuracy and generalisability of the model. Inspired by text classification research, this chapter explores four feature selection approaches to evaluate aspect usefulness: Information Gain, Chi-squared test, document frequency and the proposed aspect weighting approach in Chapter 5.

7.2 Aspect Selection

Algorithms used to rank aspects for selection can be categorised into supervised and unsupervised approaches. Supervised approaches require class labels to measure the relevance of an aspect to a particular class. In contrast, unsupervised approaches do not require any labelled data and rely on descriptive statistics of the data to detect irrelevant features.

7.2.1 Supervised Approaches

Supervised approaches were commonly used in text classification to reduce the vocabulary with which classification models were built. Feature selection approaches such as IG and CHI measures the relevance of a term based on the frequency of the presence and absence of a term in documents of a certain class. Essentially, a term that appears much more frequently in documents of a particular class than any other class has discriminative power, i.e. it is relevant for the purpose of distinguishing between these classes. Using this principle, it can be assumed that an aspect which frequently occurs in only one class to be more useful. However the question that must be addressed here is defining a useful notion of "class" for the goal of recommendation.

A product could be given a high or low rating due to the performance of its unique aspects (e.g. optical zoom camera in smart phones), which is one of the key factors that influence purchase decisions (Gao and Cui, 2016). It is not uncommon to find unique aspects in products. This is because every product needs to provide unique value (aspects) to customers to set them apart from competition (Drummond and Ensor, 2006, Nowlis and Simonson, 1996). For example, a camera might be given a high rating by having an optical zoom camera (optical zoom cameras give a better picture quality than digital zoom cameras, and as such it is a unique aspect that only certain smart phone models own) because it gives a better picture quality. Similarly, in Laptop products, having Windows Vista as the operating system (Windows Vista is an aspect belongs to the operating system aspect category) might receive lower rating from users due to its issue with security features, performance, driver support and product activation. In the context of aspect selection, the class is the user preference (preferred and less preferred) and aspects which are more frequently observed in one class than the other class are considered relevant. This thesis aims to study this observation and the utility of supervised feature selection approaches in aspect selection using user ratings as class label as suggested in the literature (Billsus and Pazzani, 1998, Miyahara and Pazzani, 2000, Vargas-Govea et al., 2011). In Amazon, users rate products based on a 5-star scale as shown in Figure 7.2. Here, a 5 and 4 stars indicate the user loved or liked the product, 1 and 2 star(s) indicate that the user disliked the product, and a 3 star rating does not indicate a strong opinion on the product. Based on these descriptions, the numerical user ratings are transformed into two classes: preferred and less preferred. Products are labelled as preferred if the overall user rating for the product is 4.0 and above, or less preferred otherwise.

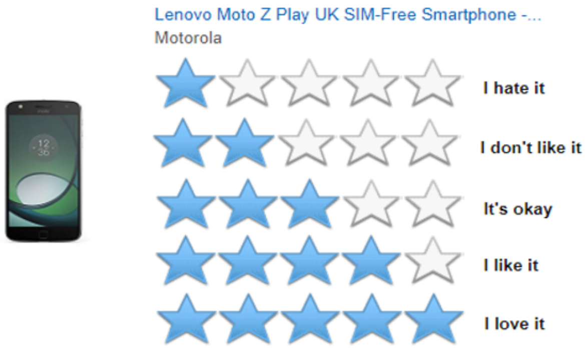


Figure 7.2: User Ratings.

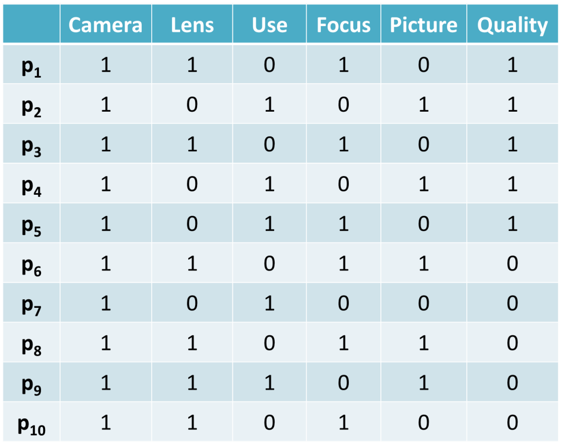
Here, a binary class is used such that c is either 0 meaning that the product is preferred by the users; or is 1 meaning it is less preferred. In this way, each product can be assigned a binary label with the following function:

$$c(p) = \begin{cases} 0, & \text{if users' ratings} \geq 4.0; \\ 1, & \text{otherwise.} \end{cases} \tag{7.1}$$

where $c(p)$ is the function that assign a class to each product.

7.2.1.1 Information Gain (IG)

Information gain has been widely used in text classification to measure the discriminative power of a term by measuring the absence and presence of a term in a document (Yang and Pedersen, 1997). In the context of social recommender systems, instances are the products and features are the aspects of a product. The aspects are scored using IG computed on their presence and absence in reviews of preferred ($c = 0$) and less preferred products ($c = 1$). For this purpose, a product p is represented as $\vec{x} = \{x_1, ...x_{|\mathcal{A}|}\}$ where x is binary valued and corresponds to the presence or absence of an aspect $a \in \mathcal{A}$. For instance, in Figure 7.3 aspect *camera*, *lens* and *focus* has a value of 1 for p_1 meaning that these aspects are present in reviews of p_1.



	Camera	Lens	Use	Focus	Picture	Quality
p_1	1	1	0	1	0	1
p_2	1	0	1	0	1	1
p_3	1	1	0	1	0	1
p_4	1	0	1	0	1	1
p_5	1	0	1	1	0	1
p_6	1	1	0	1	1	0
p_7	1	0	1	0	0	0
p_8	1	1	0	1	1	0
p_9	1	1	1	0	1	0
p_{10}	1	1	0	1	0	0

FIGURE 7.3: Aspects Vector ($\vec{x}$).

The information gain of an aspect, a, given the classes $c \in C$ is computed as follows (Mitchell, 1997):

$$\mathrm{IG}(C, a) = H(C) - H(C|a) \tag{7.2}$$

where $H(C)$ is the parent entropy and $H(C|a)$ is the weighted entropies of the children nodes that are partitioned by the absence and presence of aspect a. The application of IG in this work is illustrated using a decision stump in Figure 7.4. The internal node (parent) consists of the entire population of 120 products in a particular dataset. The left leaf node (child) is formed by 80 products in which the aspect is present and among these products, 20 of them are preferred products and the rest are less preferred products. The right leaf node (child) is formed by the remaining products in which the aspect is absent. As a result, the information gain for aspect a is 0.17. Aspects with low information gain are unlikely to have a high rank position and thus unlikely to be relevant. Therefore, this suggests that aspect a is irrelevant in product representation to distinguish preferred products from less preferred products.

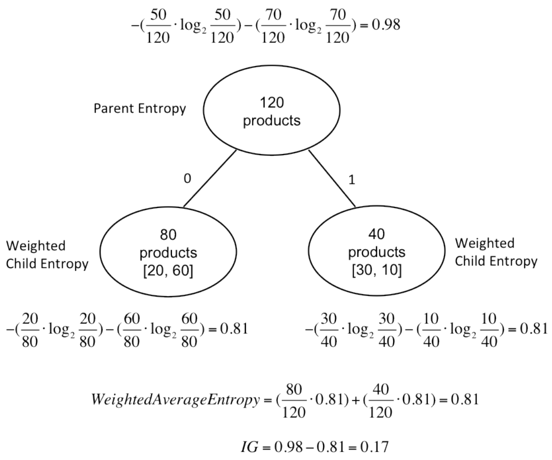

$-(\frac{50}{120}\cdot\log_2\frac{50}{120})-(\frac{70}{120}\cdot\log_2\frac{70}{120})=0.98$
Parent Entropy
120 products
0
1
Weighted Child Entropy
80 products [20, 60]
40 products [30, 10]
Weighted Child Entropy
$-(\frac{20}{80}\cdot\log_2\frac{20}{80})-(\frac{60}{80}\cdot\log_2\frac{60}{80})=0.81$
$-(\frac{30}{40}\cdot\log_2\frac{30}{40})-(\frac{10}{40}\cdot\log_2\frac{10}{40})=0.81$
$WeightedAverageEntropy=(\frac{80}{120}\cdot0.81)+(\frac{40}{120}\cdot0.81)=0.81$
$IG=0.98-0.81=0.17$

FIGURE 7.4: Decision stump for aspect a

7.2.1.2 Chi-Squared (CHI)

The Chi-squared test is a statistical test that measures the independence between two events. Here, Chi-squared is used to measure the independence between an aspect and a class. A high score on the Chi-squared test indicates that the aspect and the class are dependent and a value of 0 indicates that they are independent. The Chi-squared statistic is defined as (Yang and Pedersen, 1997):

$$\text{CHI}(a,c)=\frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \tag{7.3}$$

where N is the number of products, A is the number of times the aspect a appears in reviews of preferred products, B is the number of times that a appears in reviews of less preferred products, C is the number of preferred products that do not have a, and D is the number of less preferred products that do not have a. These notations are summarised in the contingency table in Table 7.5.

	$c=0$	$c=1$
$x=0$	A	B
$x=1$	C	D

TABLE 7.5: Presence and Absence of a in class 0 and 1

where $x=0$ when aspect a appears in the reviews of a product and $x=1$ when it is absent from the product's review. Information gain and Chi-squared are among the most effective methods of feature selection for text classification (Yang and Pedersen, 1997). One of the advantages of Chi-squared over information gain is that it is easier to compute. However, Chi-squared is not reliable for low frequency terms (Dunning, 1993). Therefore, aspects which have a low frequency may be incorrectly ranked by the Chi-squared measure.

7.2.2 Unsupervised Approaches

7.2.2.1 Preference-based Aspect Weights

Chapter 6 discussed an approach to assign weights to the extracted aspects in Equation 6.5. Unlike information gain and Chi-squared, which measure aspect relevance based on the presence and absence of an aspect in a class, the preference-based aspect weighting approach infer aspect importance using a preference graph generated from users purchase preferences and sentiment knowledge. This approach is based on the intuition that aspect importance arises when the same set of aspects contributes to similar purchase decisions. Similarly, aspects which contribute to purchase decisions are relevant in product representation. Therefore, preference-based aspect weighting is used to determine the relevance of an aspect such that aspects that were assigned higher weights are deemed relevant.

7.2.2.2 Document Frequency

Document frequency selects terms that frequently occur in product reviews. It scales well to large corpora and often does well when there are many thousands of aspects selected (Yang and Pedersen, 1997). Therefore, frequency-based selection approaches can be a good alternative to more complex methods. Accordingly, the aspects are ranked based on their DFREQ scores, computed as follows:

$$\text{DFREQ}(a) = \frac{f(a)}{\sum_{j=1}^{\mathcal{A}} f(a_j)} \tag{7.4}$$

where DFREQ(a) returns the relative frequency of an aspect a appearing in reviews $\mathcal{R}$. Here the frequent occurrence of aspects in online reviews is perceived as important and should receive a higher rank position. The key difference between supervised approaches and unsupervised frequency-based approaches is that in the former approach, aspects which appear in the reviews of preferred and less preferred products will have a lower rank position; whilst with the latter aspects which frequently appear in the reviews of products will receive a higher rank position.

7.3 Generating Recommendation

The aim of aspect selection is to reduce the size of the set of aspects $\mathcal{A}$ for product p to a smaller aspect subset size n by selecting aspects according to the score assigned by the feature selection technique. Algorithm 1 details the process to rank aspects for selection. For every a, ranking scores were calculated using a feature selection technique (step 3). Then, the aspects are ranked based on the ranking score such that aspects that ranked at the top are deemed important (step 5). Finally, the top n aspects are selected to form a corresponding reduced aspect set $\mathcal{A}'$ for product p, where $\mathcal{A}' \subset \mathcal{A}$ and $|\mathcal{A}'| \leq |\mathcal{A}|$.

Algorithm 1 Aspect Selection

```
Input:   $\mathcal{A}$, Set of aspects extracted from product training set
Output:  $\mathcal{A}'$, Set of selected aspects
         $FS$, Feature selection technique
1: $score = \emptyset$
2: for each $a \in \mathcal{A}$ do
3:     $score \leftarrow$ applyFS$\{a\}$
4: end for
5: Rank aspects based on ranking scores
6: $\mathcal{A}' \leftarrow score.select\{n\}$                ▷ select top $n$ aspects
7: return $\mathcal{A}'$
```

The set of selected aspects is used to compute *ProductScore* discussed in Chapter 5. Given a query product Q, the *ProductScore* of a candidate product, p_i, is computed using the set of selected aspects. Accordingly, the following is the reformulation of $ProductScore$.

$$ProductScore(p_i, a_j) = \frac{\sum_{j=1}^{|\mathcal{A}'|} AspectSentiScore(p_i, a_j)}{|\mathcal{A}'|} \tag{7.5}$$

The subset of aspects $\mathcal{A}'$ in Equation 7.5 considers all aspects as equally important. However, relevant aspects may not be equally important to users. Therefore, the *ProductScore* in Equation 6.1 is re-formalised by replacing $\mathcal{A}$ with $\mathcal{A}'$ as follows:

$$ProductScore(p_i, a_j) = \frac{\sum_{j=1}^{|\mathcal{A}'|} AspectWeight(a_j) * AspectSentiScore(p_i, a_j)}{\sum_{j=1}^{|\mathcal{A}'|} AspectWeight(a_j)} \tag{7.6}$$

7.4 Evaluation of Aspect Selection

The aim of this evaluation is to assess the effect of aspect selection on recommendation performance. First, an experiment was conducted to ascertain the performance of each feature selection technique in aspect selection by comparing their performance to an approach that does not use aspect selection (AllAspects). The comparative study includes the following approaches:

- AllAspects - Aspect extraction approach, DirectRelations$^+$, introduced in Chapter 5.
- AS+IG - Information Gain (see Section 7.2.1.1).
- AS+CHI - Chi-Squared statistics (see Section 7.2.1.2).
- AS+Pref+ST - Uses preference-based aspect weights with sentiment threshold (see Section 7.2.2.1).
- AS+DFREQ - Document Frequency (see Section 7.2.2.2).

In this experiment, the standard feature selection experimental methodologies were used where selection heuristics are applied to the training set and the top n aspects are selected and graphed against the results of MAP and RI (Wiratunga et al., 2004, Yang and Pedersen, 1997). Thereafter, the effect of assigning aspect weights on the selected aspects were studied by applying the aspect weighting approach introduced in Chapter 6 on the selected aspects.

7.4.1 Comparison of Feature Selection Techniques

Figure 7.5 to 7.18 shows the results in terms of MAP and RI respectively on the seven datasets. For each approach, the graph shows the results computed at different aspect subset sizes. In general, these graphs shows that AS-based approaches can achieve better results over AllAspects for certain aspect subset sizes. Specifically, results show that in majority of the datasets (4 out of 7) AS+DFREQ outperforms the rest in MAP and RI. The finding from these results confirms observations from Zhang et al. (2010b) who also observed frequency-based approaches leading to better performance.

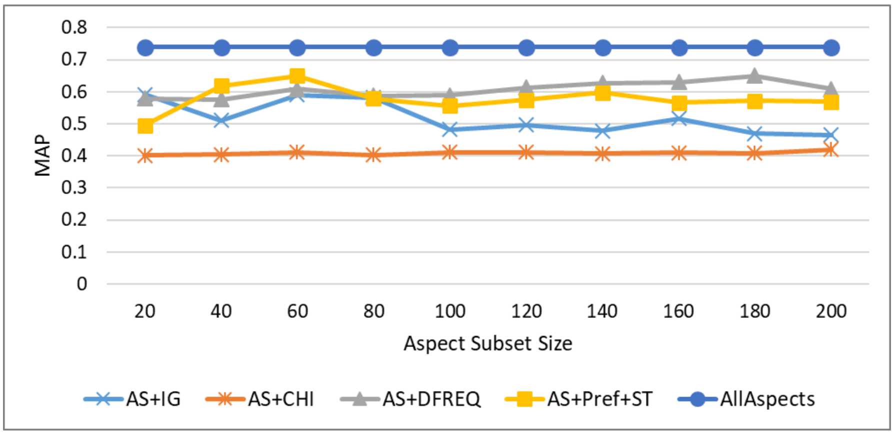


Figure 7.5: MAP for DSLR

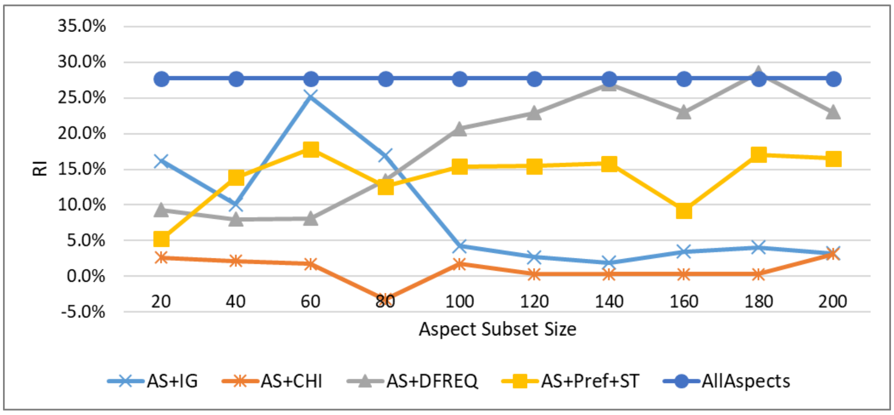


Figure 7.6: RI for DSLR

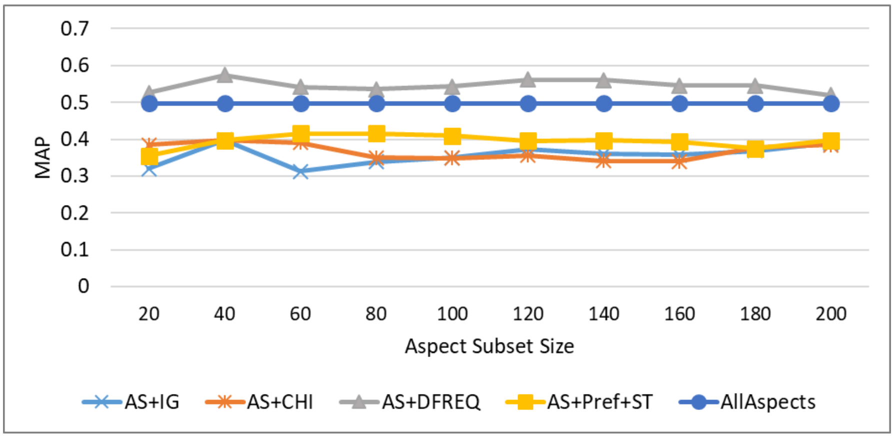


FIGURE 7.7: MAP for Laptops

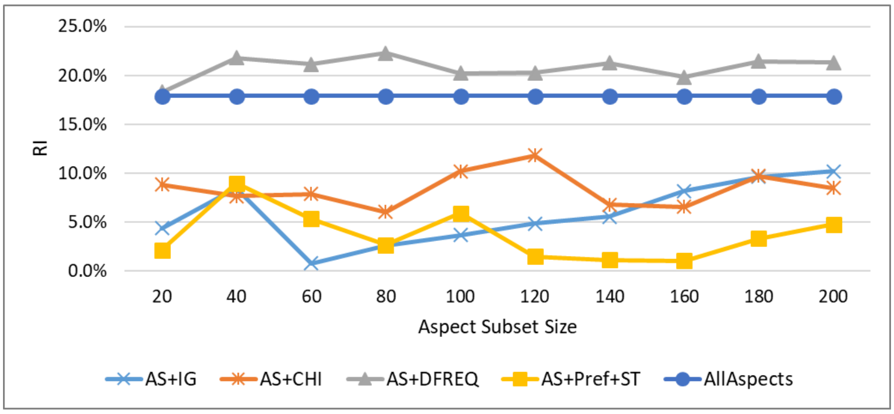


FIGURE 7.8: RI for Laptops

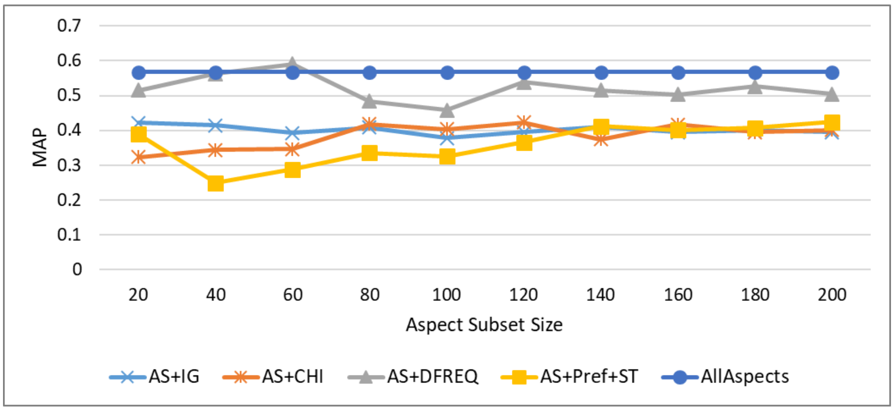


FIGURE 7.9: MAP for Tablets

It can be observed that Laptops benefit most from the AS+DFREQ approach, where improvement is observed across different aspect subset sizes in both MAP and RI (Figure 7.7 and 7.8). In TV, it can be observed that the best performing technique is AS+Pref+ST in MAP and RI (see Figures 7.17 and 7.18). Specifically, it can be observed that for a smaller aspect subset size, AS+Pref+ST gives better recommendation

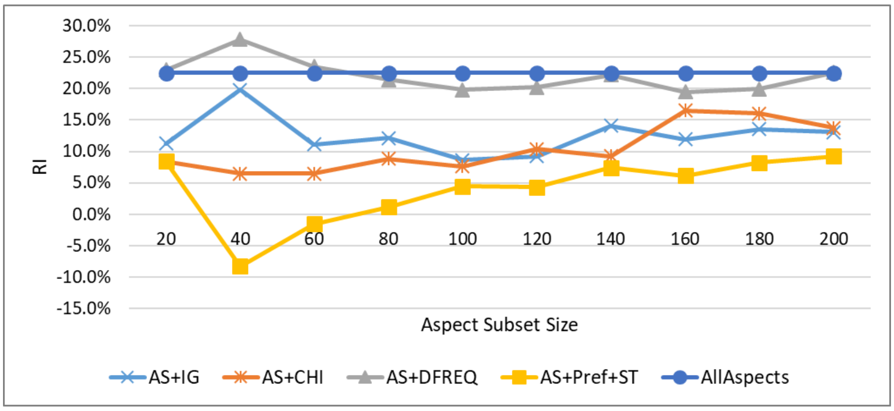


FIGURE 7.10: RI for Tablets

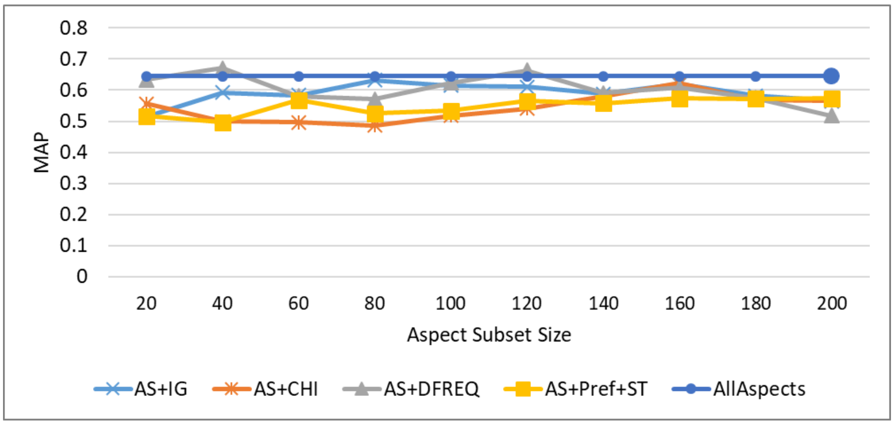


FIGURE 7.11: MAP for Phones

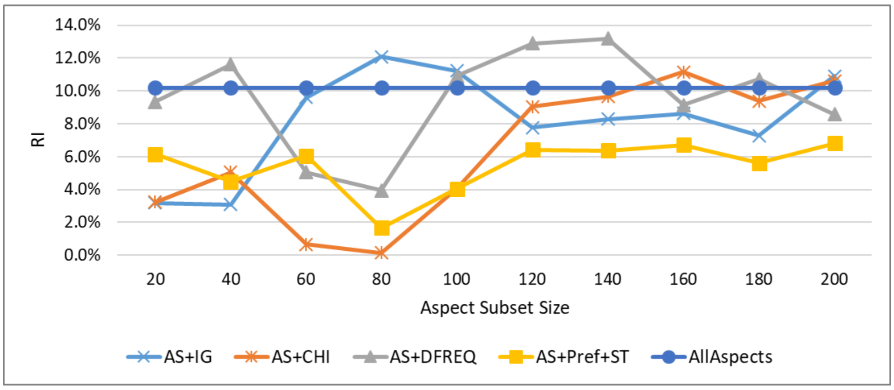


FIGURE 7.12: RI for Phones

performance in both MAP and RI, but as the aspect subset size is greater than 40, its recommendation performance falls below AllAspects. AS+DFREQ also achieves similar performance to AS+Pref+ST in RI at the aspect subset size of 20 where AS+DFREQ and AS+Pref+ST achieves RI of 6.7% and 7.3% respectively. Similarly, at the size of 40, AS+DFREQ and AS+Pref+ST achieves RI of 8.2% and 9.0% respectively. The results in TV suggest that the performance of AS+DFREQ is comparable to AS+Pref+ST.

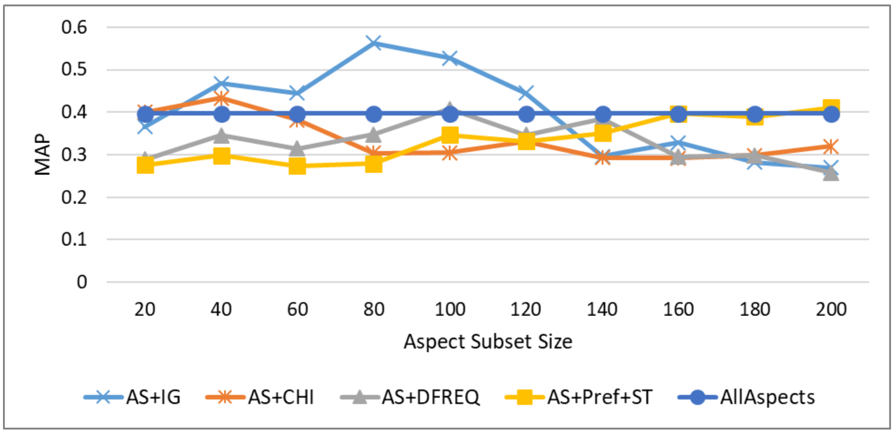


FIGURE 7.13: MAP for Printers

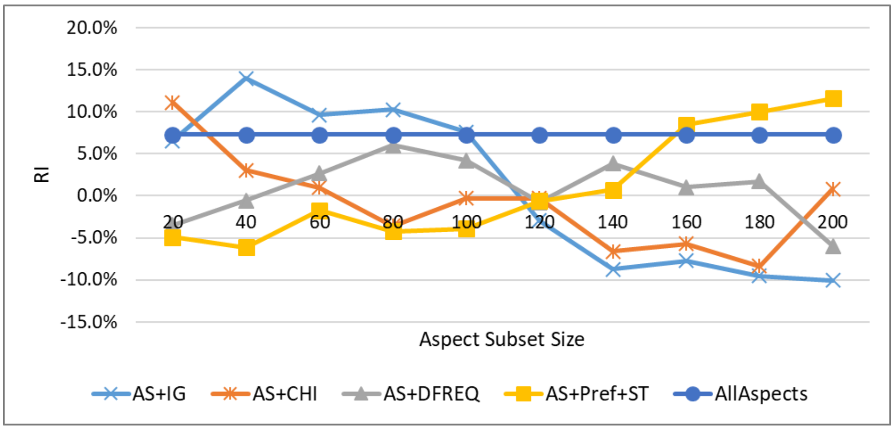


FIGURE 7.14: RI for Printers

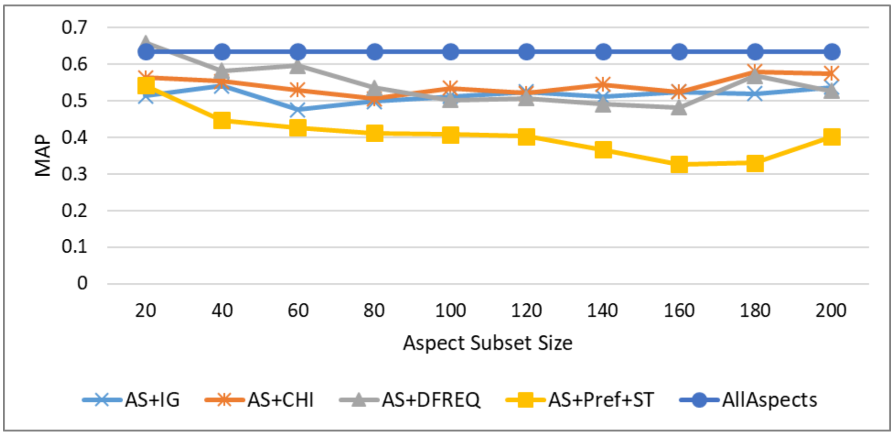


FIGURE 7.15: MAP for Mp3

Similar observations can be made in the Tablets, Phones and Mp3 datasets in Figures 7.9 to 7.12 and Figures 7.15 to 7.16 where AS+DFREQ approach performs better than AllAspects approach in both MAP and RI when the aspect subset size is at 60 or below. Specifically, in the Tablets, Phones and Mp3 datasets, small improvements can

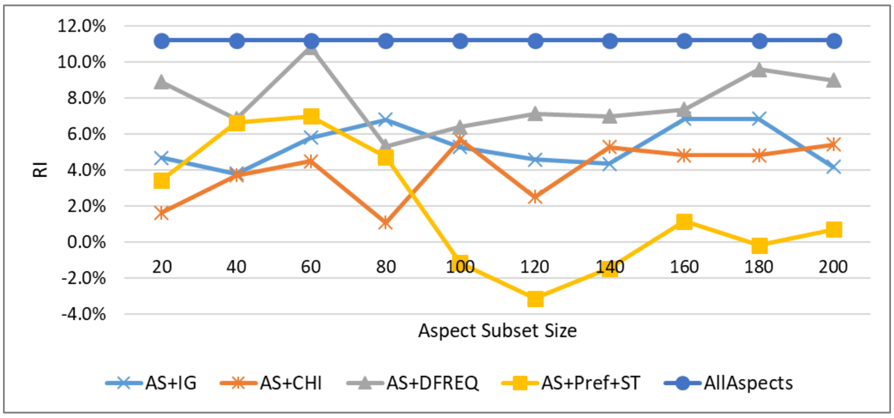


FIGURE 7.16: RI for Mp3

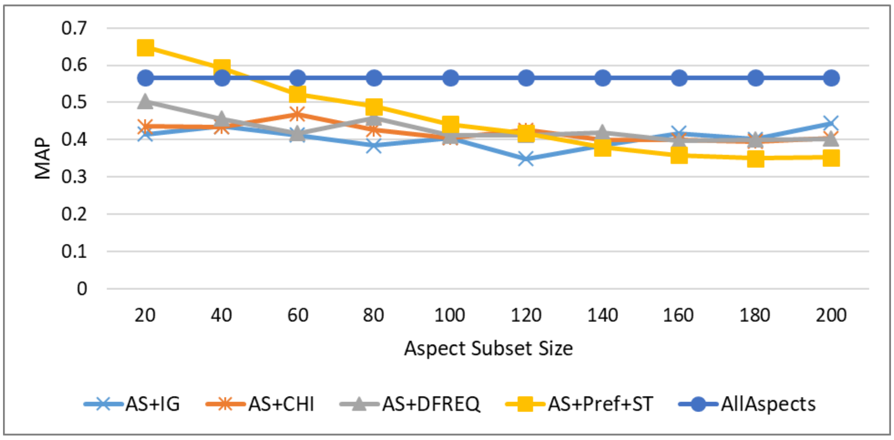


FIGURE 7.17: MAP for TV

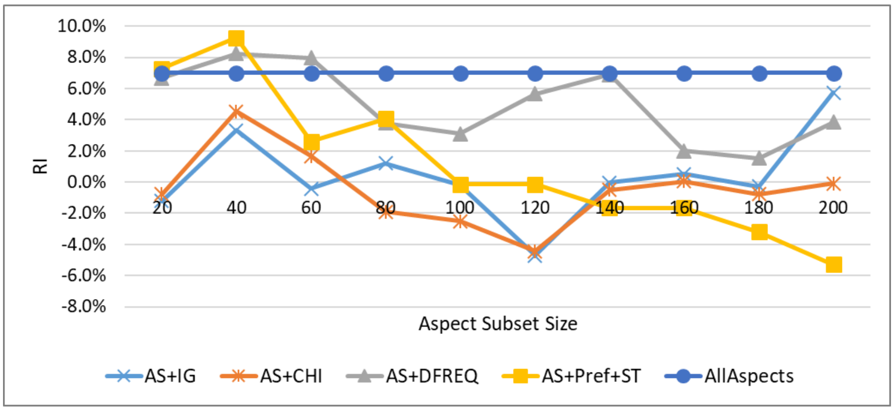


FIGURE 7.18: RI for TV

be observed in the MAP metric at aspect subset sizes of 60, 40 and 20 respectively. In RI, best results in Tablets is at aspect subset size of 40. In Phones, AS+DFREQ performs better than AllAspects at aspect subset size of 40. However, the best results achieved by AS+DFREQ is at aspect subset size of 140 with a RI of 13%. In Mp3, AS+DFREQ achieve the same results as AllAspects at 60. Here, AS+DFREQ performs better in

Tablets when the aspect subset size is bigger and in Mp3, AS+DFREQ performance is similar to AllAspects with a smaller subset size. This behaviour could be explained by looking at the list of aspects ranked by AS+DFREQ obtained from one of the folds in Table 7.6. It can be observed that the same aspect can be referred to using alternative vocabulary. For instance, in Phones, aspects *battery life* and *charge* refer to the aspect *battery*. In addition, reviewers also refer to the product they bought in different ways. For example, tablet is also referred to as *item* or *product*, or it can be referred to based on the model of the tablet such as *iPad*[2]. Retaining the single-term aspects that are part of multi-word aspects further contributes to increase in variation of the number of aspects in the vocabulary (e.g. *picture* vs. *picture quality*). However, removing the single term aspects runs the risk of losing important aspects for the recommender system. Therefore, selection heuristics are required to determine whether these single-term aspects should be removed during aspect extraction step. In WordNet[3], *charge* does not appear as a synonym to *battery* but *battery* has a 'used for' relationship with *charge* in ConceptNet[4]. This suggests that *charge* and *battery* is semantically similar. Therefore, in order to reduce the aspect subset size, clustering aspects based on the concepts of the electronic products is needed.

One possible reason for the aspect subset size for Mp3 players to be smaller than Tablets and Phones is due to the increasing functionality of smart phones and tablets in recent developments of those products. For example, both smart phones and tablets can now play mp3 files. As a result, there are more aspects that reviewers can comment on which leads to a larger vocabulary size used by the reviewers to comment in their reviews. Therefore, it is reasonable to assume that the number of aspects mentioned in Mp3 players are fewer than Phones and Tablets.

Figures 7.13 and 7.14 show that AS+IG and AS+CHI perform better than AllAspects and the unsupervised feature selection approaches in Printers. In MAP, AS+IG and AS+CHI achieved best results at aspects subset size of 80 and 40 respectively. In

[2] As mentioned earlier in page 119, product category such as'*tablet*' is a general aspect which users used to build the overall opinion on the product.

[3] A lexical database for English language that can be used to find synonyms. It is available at https://wordnet.princeton.edu

[4] ConceptNet is a freely-available semantic network, designed to help computers understand the meanings of words that people use. It is available at http://conceptnet.io

DSLR	*Laptops*	*Tablets*	*Phones*	*Printers*	*Mp3Players*	*TV*
camera	laptop	tablet	phone	printer	player	picture
lens	screen	screen	screen	print	music	sound
canon	work	ipad	battery	work	work	quality
picture	price	work	camera	ink	mp3	price
quality	keyboard	app	app	set	product	work
video	machine	price	work	printing	review	set
nikon	window	product	price	paper	sound	screen
shoot	problem	quality	android	quality	mp3 player	purchase
focus	computer	battery	quality	set up	device	samsung
image	make	problem	battery life	cartridge	play	remote
make	quality	money	call	wireless	quality	watch
dslr	battery	play	iphone	color	song	amazon
feature	touch	android	feature	problem	screen	problem
photo	drive	device	device	scan	button	color
shot	run	purchase	charge	purchase	price	picture quality
light	product	game	size	canon	listen	room
work	purchase	back	card	price	charge	size
iso	feel	set	google	page	video	box
screen	light	size	money	make	mp4	cable
mode	money	review	nexus	setup	battery	app
feel	start	charge	samsung	product	card	product
price	lenovo	life	update	back	exchange	smart tv
body	mouse	samsung	hand	document	radio	feature
purchase	amazon	watch	set	black	ipod	television
point	sound	apple	run	computer	file	roku
upgrade	review	download	play	photo	mp4 player	sony
review	life	camera	sound	install	feature	review
photography	acer	battery life	light	brother	item	model
setting	set	lot	case	support	purchase	black
flash	pad	read	review	feature	headphone	setting

Table 7.6: Top 30 Aspects Ranking Using AS+DFREQ

RI, the best results is observed with a smaller aspects subset size in both approaches. Specifically, the RI for AS+IG is 14% at size 40 and the RI for AS+CHI is 11% at size 20. The proposed supervised feature selection approach that uses user ratings as the criteria to generate class labels improves recommendation performance in Printers and performs poorly in other datasets. This can be explained by analysing the user ratings distribution for each dataset. In Figure 7.19, it can be seen that at least 75% of the user ratings in DSLR, Tablets, Phones and TV are greater than 4.0. Further, more than 50% of the products in Mp3 and Laptops dataset have ratings of less than 4.0. In contrast, Printers has a relatively symmetrical distribution compared to other datasets, with ratings that are between 3.2 to 4.7. Given that the class label was assigned based on the user rating of 4.0, where products with a rating ≥4.0 are labelled as preferred and <4.0 as less preferred, an analysis of the boxplot suggests that all datasets except Printers suffer from the class imbalance problem where these datasets are either have majority of the product labelled as preferred products or less preferred products. As a result, this limits the performance of AS+IG and AS+CHI. The finding from these results confirms previous findings that the performance of feature selection technique degrades when there is a class imbalance problem (Forman, 2003). However, due to the class imbalance problem that occurs in a majority of the datasets, there is insufficient evidence to show that supervised feature selection approaches using user ratings as class labels are effective in improving recommendation performance. Therefore, future work is required to test this assumption.

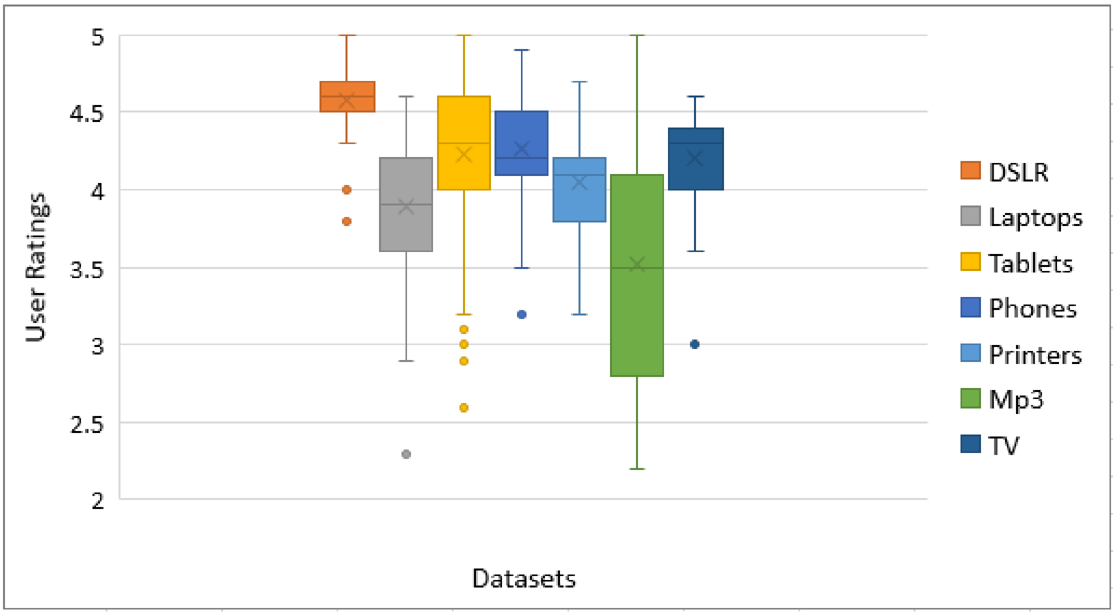


Figure 7.19: User Ratings Distribution for all Datasets

Based on the results in Figures 7.5 and 7.6, feature selection techniques do not improve recommendation performance for DSLR. The supervised approaches are unable to perform well due to 90% of the products being labelled as preferred products, this leaves only 10% of the products labelled as less preferred as can be seen in Figure 7.19. Although there are no improvements observed in MAP, the RI results show that at aspect subset size 180, AS+DFREQ achieves similar results to AllAspects. Given that in AllAspects, a total of 4,298 aspects were used to generate recommendations, AS+DFREQ has only used 4.2% of the aspects to achieve similar results. This demonstrates the benefits of aspect selection where similar results can be achieved with significantly fewer number of selected aspects. The aspect subset size in DSLR (e.g. 180) is relatively large compared to other datasets. It can be observed from the statistics in Table 4.1 (see Chapter 4) that on average the DSLR reviews are longer. This suggests that lengthy reviews in DSLR camera products are likely to have a greater descriptive vocabulary for aspects.

7.4.2 Analysis on Recommendation Performance with Selected Aspect Weights

The experiment on aspect selection is extended by comparing the results of aspect selection (AS) with and without aspect weights (AS#). Since the best aspect subset size on the majority of the datasets observed in the previous experiment was between 20 to 80, a median value of 50 is chosen for comparison here. Figures 7.20 to 7.27 show the comparison results of the four feature selection techniques for each dataset. It can be observed that there is little or no improvement in MAP for all datasets after applying aspect weights following selection. In the RI results, it can be observed that weights do not benefit AS+IG and AS+CHI. In Figure 7.21, it can be seen that applying weights on selected aspects ranked by IG only improves RI on Mp3. Although there is an improvement observed with Laptops, the RI achieved by AS#+IG is only 5.1%[5] which means this approach is not able to recommend '*better*' products. Furthermore, negative

[5] 5.1% is not ideal because it is less than 6.7%. Recall in Chapter 5 that a minimum RI of 6.7% is required in order to show that the algorithm is recommending products that are at least one rank position higher ('*better*') than the query product.

RI was observed when weighting is applied in TV with AS#+IG. This also shows that AS#+IG is not able to recommend '*better*' products in TV.

Figure 7.23 shows the results for AS+CHI and AS#+CHI. Here, DSLR benefits most from weighting, followed by Tablets and Mp3. However, it can be seen that Laptops, Phones, Printers and TV perform poorly with weighting. Recall from the previous experiments in Section 7.4.1 that Printer benefited most from AS+IG and AS+CHI. Here, the results for Printer has dropped after applying weighting. This further confirms the negative influence of weighting aspects that are selected by supervised feature selection techniques.

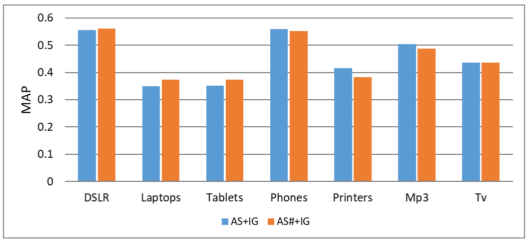


Figure 7.20: MAP for IG.

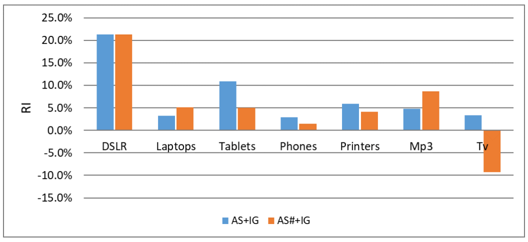


Figure 7.21: RI for IG.

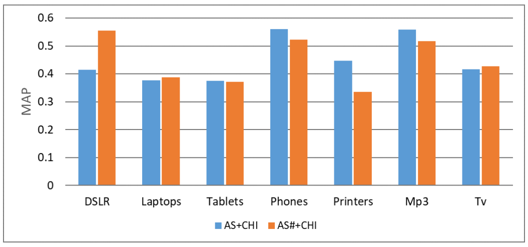


Figure 7.22: MAP for CHI.

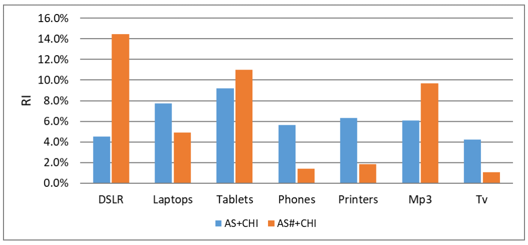


Figure 7.23: RI for CHI.

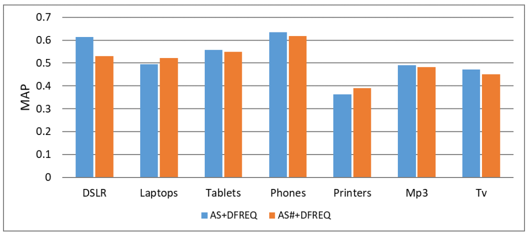


Figure 7.24: MAP for DFREQ.

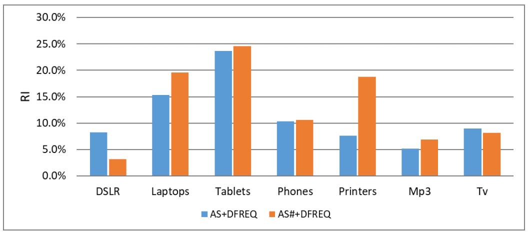


Figure 7.25: RI for DFREQ.

In contrast, the RI results on unsupervised feature selection techniques show improvement across all datasets. Specifically, AS#+DFREQ improves RI in Laptops, Tablets, Phones, Printers and Mp3 by at least 4% while AS#+Pref+ST improves on all datasets except Laptops. It can be observed that AS#+Pref+ST benefits the most with at least

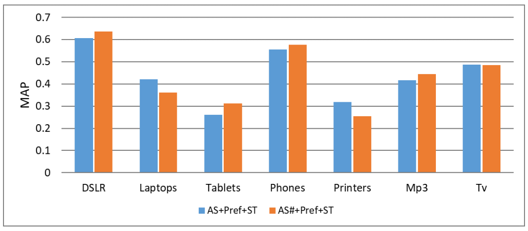


Figure 7.26: MAP for Pref+ST.

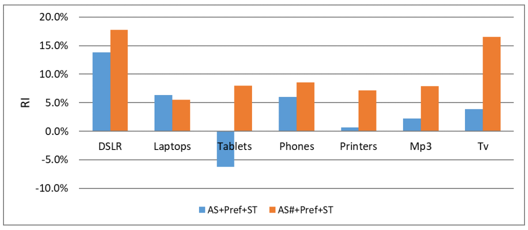


Figure 7.27: RI for Pref+ST.

20% performance improvement observed in Tablets, Printer, Mp3 and TV. These results suggest that DFREQ and Pref+ST alone is not an optimal approach for aspect selection. Overall it can be observed that RI performance on unsupervised approaches improved with aspect weighting, demonstrating the advantage of aspect selection with aspect weights for recommendation algorithms when unsupervised feature selection approaches are used.

7.5 Chapter Summary

The high dimensionality of data due to aspect extraction has a detrimental affect on recommendation performance. However, aspects extracted are not equally useful in product representation and therefore are subjected to selection-based dimensionality reduction. This chapter demonstrates how feature selection techniques from machine learning research can be exploited for this purpose. Furthermore, to address the issue of the absence of class labels, the use of user ratings as class label proxies was introduced. The decision was motivated by the availability of users ratings and research on consumer behaviour. This is useful particularly when adopting supervised feature selection methods commonly reported in the literature such as information gain and Chi-squared. Besides the frequency-based feature selection technique, a preference-based feature selection method to rank aspects where each aspect is ranked using a preference graph and sentiment knowledge was also proposed. Further, considering relevant aspects are not equally important to users, a ranking model was formalised by weighting the selected aspects in product representation.

The evaluation results demonstrated that aspect selection improves recommendation performance. A comparative study of the four feature selection techniques suggests that the unsupervised feature selection approach DFREQ performs best in majority of the datasets. The performance of supervised approaches such as IG and CHI was limited due to the class imbalance problem. However, in the absence of class imbalance problem, the results demonstrated that supervised approaches perform better than the unsupervised approaches. An analysis of the difference in aspect subset size shows that users use different terms to refer to the same aspect. Therefore, in order to reduce the number of aspects, the semantic similarity between aspects needs to be considered.

The aspect selection experiment was extended to assess the effect of applying the aspect weights on the selected aspects. Based on the results of the experiment in Section 7.4.2, aspect weights benefit unsupervised approaches (AS+DFREQ and AS+Pref+ST). When using a supervised approach, recommendation performance is better without integrating the aspect weights.

Chapter 8

Conclusions

This thesis addresses the limitations of using user-generated reviews to enhance recommendation performance. To overcome these limitations,the following research questions were set out in the context of social recommender systems:

- Which dependency relations are most relevant to extract aspects that improve recommendation performance?
- Which explicit and implicit knowledge sources can be integrated and what impact do they have on recommendation performance?
- Can feature selection methods used for dimensionality reduction in classification be used to select relevant aspects for social recommendation?

In order to address these questions four objectives were identified. This chapter discusses the contributions of this thesis by revisiting these initial research objectives and summarising key conclusions that emerged from the experimental studies in relation to each research objective. This chapter also present future directions and desirable extensions, before finally concluding this thesis.

8.1 Objectives Revisited

1. **Develop a dependency-based product aspect extraction technique that improves recommendation performance.** In this research, aspects extracted from online product reviews are used to represent products. In Chapter 5, a dependency rule-based aspect extraction approach, DirectRelations^{+}, that combines the strengths of both dependency relations and rule-based frequent noun approaches was presented. To address the first research question, this work shows how dependency rules are selected and the benefits of combining them with a rule-based frequent noun approach. The utility of the proposed aspect extraction algorithm is evaluated in a recommendation setting with existing state-of-the-art dependency-based approaches. The results show that combining a rule-based frequent noun approach with an informed selection of dependency relations achieves overall best results. Specifically, using the rule-based frequent noun approach to filter spurious aspects and apply sentiment knowledge to select dependency relations is crucial in extracting meaningful aspects for product recommendation. However, it has been observed that the rule-based frequent noun approach does not benefit large datasets. This is because this approach is not able to recognise the different ways that reviewers used to refer to the same aspect. Hence, some of the important aspects are removed by this approach.

2. **Develop a product ranking algorithm using social knowledge captured from product reviews and users purchase preferences.** Aspects that are likely to have influenced the users' purchase decisions are deemed important. To address the second research question, Chapter 6 presented a novel aspect weighted sentiment scoring approach to rank products. This is achieved by first inferring aspect importance weights from users purchase preferences represented in a preference graph and subsequently integrating social knowledge from product reviews (explicit feedback) and users' purchase preferences (implicit feedback). Evaluation results show that recommendation performance benefits from the aspect weighted sentiment-driven approach compared to baseline approaches which do not consider aspect weights (Cosine, BetterScore and PageRank). This demonstrates that users

purchase preferences is a promising source of implicit knowledge in estimating aspect importance. Specifically, to achieve the significant benefit of using a preference graph to infer aspect importance, there is a need to take into consideration that reviewers who voted strongly in favour of an aspect might overpower those of others, who might have a less strong opinion. Therefore, aspect weights are estimated using the polarity of a sentiment score (positive and negative) for each view-purchased product pair instead of sentiment scores. Experimental results provide evidence that setting a sentiment threshold to determine the polarity of an aspect when comparing aspect sentiment values in every view-purchase product pair provides significant benefits to recommendation performance.

The distribution of sentiments of an aspect is considered when generating aspect weights. To do this, Gini and Wilson score were applied. Evaluation results show that Gini scores does not improve recommendation performance due to little or no social agreement among the reviewers were found on the sentiment expressed over the aspects. Thus, Gini has little effect on recommendation performance. In contrast, results for Wilson score is mixed where better performance is observed in specific datasets. Further analysis on the Wilson scores in the dataset shows that the limited occurrence of unique aspects limits the opportunity for Wilson score to improve recommendation performance.

In this research, products are represented using product aspects extracted from reviews. Therefore, the recommendation performance is also affected by the quality of the aspects. Evaluation results show that having many shared aspects between products has a detrimental effect on recommendation performance. This is not surprising as a higher number of shared aspects will increase the opportunity of using spurious aspects to represent products. Therefore, this motivates the need to explore selection strategies to identify a subset of relevant aspects for product representation.

3. **Investigate the utility of feature selection techniques to select relevant aspects for product representation.** Chapter 7 presented four supervised and unsupervised feature selection techniques to address the third research question.

For supervised methods, an approach to define class labels for products using users ratings was presented. A comparative study of four feature selection technique show that the unsupervised feature selection approach, DFREQ gives the best performance. The performance of supervised approaches such as IG and CHI is poor in datasets which have class imbalance problem. However, in the absence of class imbalance problem, the results demonstrate that supervised approaches perform better than unsupervised approaches. It is important to point out that, there is only one dataset which demonstrate improved recommendation performance in a supervised feature selection approach. Thus, further experiments are required to ascertain the advantage of using users ratings as class labels for supervised feature selection.

An analysis of the aspect subset size shows that users used a different vocabulary to refer to the same aspect. Future work needs to consider semantic similarity between aspects in order to reduce the aspect subset size. Further experiments were conducted to assess the effect of applying aspect weighting on the selected aspects. Results show that the aspect weighting benefits most to unsupervised feature selection approaches while in supervised approaches, recommendation performance tends to be better without considering the aspect weights.

4. **Conduct a comprehensive evaluation of all developed strategies.** An evaluation of each strategy was presented in Chapters 5, 6 and 7. The multiple algorithms comparison results obtained from the seven datasets were tested for statistical significance using a non-parametric test because it assumes neither normal distributions nor homogeneity of variance and as such is more robust and better suited to the data. The proposed approaches were empirically evaluated with the objective to recommend products that are '*better*' than a given query product. Specifically, Mean Average Precision (MAP) and Rank Improvement (RI) were used to evaluate recommendation performance in the experiments with seven real-world datasets. In this research, MAP is the accuracy metric to measure recommendation performance by comparing the algorithm's prediction against the gold standard of '*better*' products. However, evaluation on recommender systems should not only focus on accuracy. It is also important to consider how much a

recommended product is '*better*' than a query product. For instance, when users are browsing products in an e-commerce website, a list of recommended products are generated for each browsed product. In such situation, users may not be interested to see the list of recommended products with lower rating than the product the user is currently looking at (query product). Therefore, both MAP and RI are useful to evaluate recommendation performance from the perspective of system accuracy and its capability to recommend '*better*' products. The best performing algorithm should ideally achieve the best results in both MAP and RI, but a lower RI is more damaging to recommendation performance as users are likely to feel disappointed with a recommendation if they are recommended with products with a lower rating than the one they are looking at. Therefore, RI is prioritised over MAP in this thesis.

5. **Create a dataset consisting of product details from multiple product categories (Cameras, Laptops, Tablets, Phones, Printers, Mp3 players and TV) and the corresponding users' purchase preferences.** In order to estimate user preferences and evaluate recommender system, a collection of 7 product datasets has been created. Each dataset contains at least 50 products and more than 3,000 reviews. For every product, the dataset contains the product name, price, date of product release, product reviews, user ratings, best seller rank and the list of products that other consumers bought after viewing the product. A major contribution in this collection is that it contains users' purchase preferences, which allow products to be compared based on consumer's purchases. At the time of this thesis, there were no public datasets containing this information.

8.2 Future Work

This thesis focuses on the use of social knowledge to improve recommendation performance. Specifically, this thesis addresses the problem of capitalising on textual features in product reviews for social recommender systems. This section highlights some of the limitations of the work presented in this thesis and indicates future extensions.

8.2.1 Extending Aspect Extraction Approach

The heuristic rules in the proposed aspect extraction approach assume aspects are nouns that are connected with sentiment words that appear as adjectives, verbs or adverbs. However, this assumption may not hold well if extracting aspects from textual information gathered from other domains such as medical insurance claims. In the proposed aspect extraction approach, aspects extracted from electronic product reviews are assumed to be either unigrams or bigrams. However in medical claims, aspects extracted from textual information are types of accidents and trauma which are descriptive in nature (Popowich, 2005). For instance, 'fell down three flight of stairs' or 'left arm is swelling but no breaks' should map to the concept 'accident and trauma'. However, in the current approach, 'flight', 'stairs', 'arm' and 'breaks' will be extracted as aspects. Here, the mapping between aspects to a concept is not available. Therefore, future work may examine how to improve existing heuristic rules to tailor to other domains such as medical insurance.

8.2.2 Explore Different Sentiment Classification Algorithms

While this thesis investigated how different aspect extraction approaches affect recommendation performance, it underlines the need for an investigation in the effect of different sentiment analysis algorithms. Previous work shows that including emotion features improves sentiment classification accuracy in social media data (e.g. Digg, MySpace and Twitter) (Muhammad, 2016). Because of how close social media language has become to the terminology used in product reviews, it is reasonable to question whether including emotion features in sentiment classification algorithms would also help to improve product recommendations.

8.2.3 Personalised Recommendations

The recommendation approach used throughout this research is a non-personalised recommendation where products recommended to users are '*better*' than the user's query product. While a '*better*' product is favourable to the users, individual preferences should be taken into consideration in order to provide most relevant recommendations. Therefore, a natural extension to this work is to build user profiles based on their written reviews and investigate whether preference knowledge gathered from users' purchase preferences can be combined with individual user preferences to provide personalised recommendation. However, due to privacy concerns with respect to user data (e.g. personal data, browsing history), it is not easily available for research purpose.

8.2.4 Alternative Approach to Learning Aspect Weights

Another promising extension from this work is to capture aspect importance that changes over time. Previous work has demonstrated that incorporating temporal information to infer aspect weights improves recommendation performance (Ferrer et al., 2014). Therefore, it will be interesting to investigate methods to capture the evolution of user's preferences over time.

Finally, in this work aspect weights were learned by comparing sentiment difference between view-purchased product pairs. An alternative approach would be to model the recommendation problem as a simple relational classification problem, where the instances being classified are the relations between pairs of products, and the labels are the orderings of those products in the recommendation. Doing so and representing products as bags of aspects would allow for the inference of optimal aspect weights using a simple computational graph and the backpropagation of an error computed from the true ordering of the pair of products being classified. By feeding pairs of products into the network and trying to predict the ordering (either product 1 $\succ$ product 2, or product 2 $\succ$ product 1), the weights learned in the process of backpropagation could be used as optimal aspect weights.

Appendix A

SmartSA

A.1 SmartSA

SmartSA is a state-of-the-art sentiment classification system (Muhammad et al., 2016) for social media text such as reviews and microblogs. This thesis capitalised on relevant product aspects and user's sentiments expressed in product reviews for recommendation. Therefore, this thesis applied SmartSA to determine the sentiment score of sentiment-bearing words in product reviews. SmartSA leverages rich sentiment information in SentiWordNet (Esuli and Sebastiani, 2006) for contextual analysis. SentiWordNet is a sentiment lexicon that possesses a high coverage of 28,431 unique sentiment-bearing terms and each term is associated with multiple senses. Word senses for terms are ordered according to their natural usage frequency, with the first sense being the most likely to occur in a document than any other senses. Thus, the first sense can be representative for the term. Given that sentiment scores are associated to word senses in SentiWordNet, SmartSA applies word sense disambiguation to determine the right sense for the target term and extract the sentiment scores.

The extracted scores are adjusted to take into consideration of valence shifters (also called sentiment shifters). Valence shifters are words and phrases that can change sentiment orientation. There are two types of modifiers that affect term polarity: lexical and non-lexical valence shifters. Lexical valence shifters are words that are in dictionary recognisable form whereas non-lexical shifters are artificial symbols that affect the expression of sentiment such as emoticons.

A.1.1 Lexical Valence Shifters

Negation. The most common negation terms are not, never and cannot. In SmartSA, negation detection is based on a list of extended negation terms in order to handle situations where the apostrophe is omitted or misplaced (e.g. "wouldnt" and "dont"). Negation terms are sentiment-bearing. Therefore, the sentiment score of the negation term is included in the aggregation.

Intensification/Diminishing. Intensifiers and diminishers are linguistic terms that serve to increase or decrease emotional valence of the sentiment word they modify. Some examples are *very*, *really* and *extremely.* In SmartSA, sentiment words that are within the scope of an intensifier are increased (or decreased with diminishers) according to the strength of the intensifier (or diminisher).

Discourse Structure. Discourse structure is concerned with how text is organised to convey meaning. The text structure is determined through the identification of discourse segments of text, their structural arrangement and the relation among them.

A.1.2 Non-lexical Valence Shifters

Capitalisation. Capitalisation adjustment is applicable only if the rest of the text is lowercase. This is because the capitalisation may not be for emphasis but merely the writing style of the author. In SmartSA, words written in capital letters are treated as an intensifier.

Repeated letter/character. Repetition of the same letter is another way to put an emphasis on the sentiment. Therefore, repeated letters are consider as intensifiers. In SmartSA, when repeated letters are detected in a sequence, the repeated characters are reduced to two from the target term and the revised term is checked with SentiWordNet.

This allow us to avoid mistaking terms like "happy" for intensifiers. If the word is not found, then the repeated letter will be reduced to one. The occurrence of multiple consecutive exclamations or question marks or a mixture of both is also treated as sentiment intensification.

Emoticons. In social media text, emoticons are a common way to express sentiment. When an emoticon is detected in a sentence, the sentence is immediately assigned the score of the emoticon provided in the emoticons list in Thelwall et al. (2010).

The final output from SmartSA is the positive and negative sentiment scores for the target sentiment word. In this research, positive and negative scores are applied to compute aspect sentiment scores of a product, which was discussed in Chapter 5.

SmartSA obtains the polarity sentiment score of sentiment-bearing words from SentiWordNet (Esuli and Sebastiani, 2006). The score will be modified to take into consideration negation terms and lexical valence shifters (e.g. intensifier and diminish terms) that can change sentiment orientation. Ideally, the dependency relation ***neg*** (a dependency relation that relates between a negation term and the word it modifies) can be used to identify negation terms. However, the informal and non-standard writing style of users in social media is not suited to the Stanford CoreNLP sentence pre-processor which is used by SmartSA. For instance, it cannot generate ***neg*** relation with the omission of apostrophe in the sentence such as *"I dont like the screen of the camera"*. One solution is to adopt a window based approach. This is because modifiers such as negation terms and valence shifters are assumed to affect terms within a specific text window (Thelwall et al., 2012). Therefore, in order to capitalise on the contextual analysis offered by SmartSA, a window based approach was adopted to extract a window of words pivoted on the target sentiment word as a document presented to the tool for sentiment scoring.

Appendix B

SenticNet Aspect Parser Rule-based Algorithm

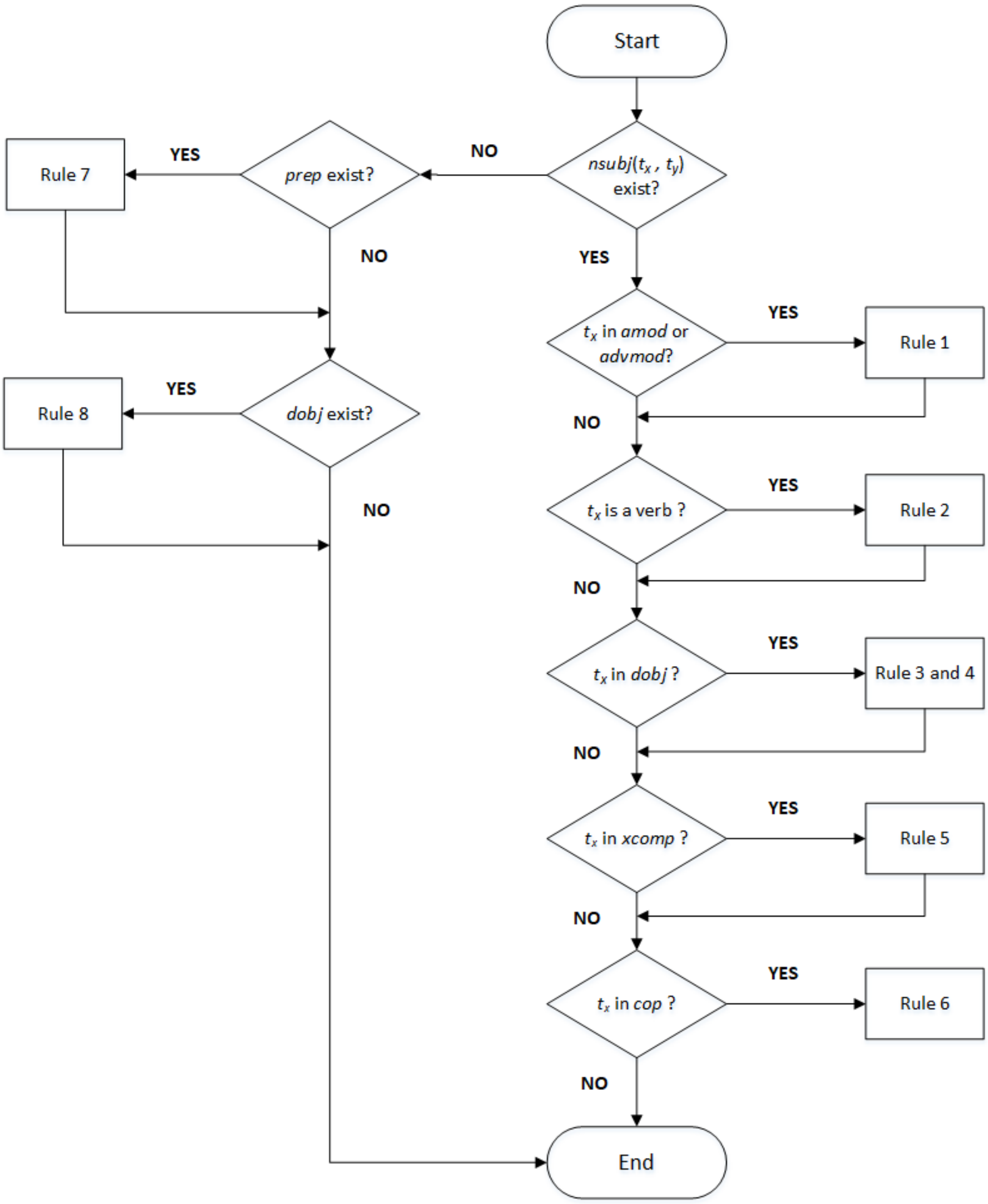


Figure B.1: Main Flowchart

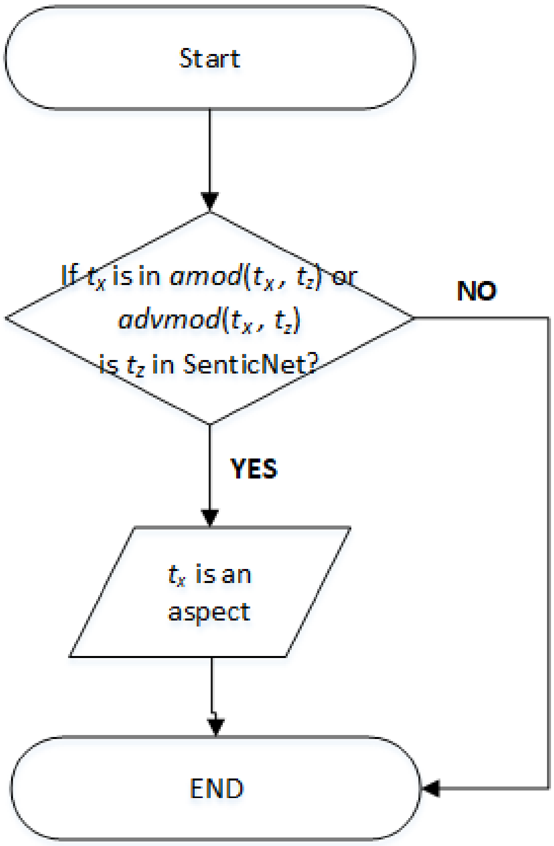

Start
If t_x is in $amod(t_x, t_z)$ or $advmod(t_x, t_z)$ is t_z in SenticNet?
NO
YES
t_x is an aspect
END

Figure B.2: Rule 1

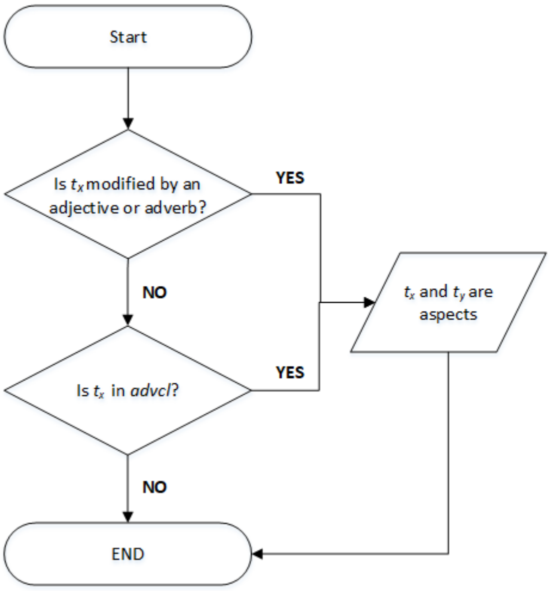

Start
Is t_x modified by an adjective or adverb?
YES
NO
t_x and t_y are aspects
YES
Is t_x in $advcl$?
NO
END

Figure B.3: Rule 2

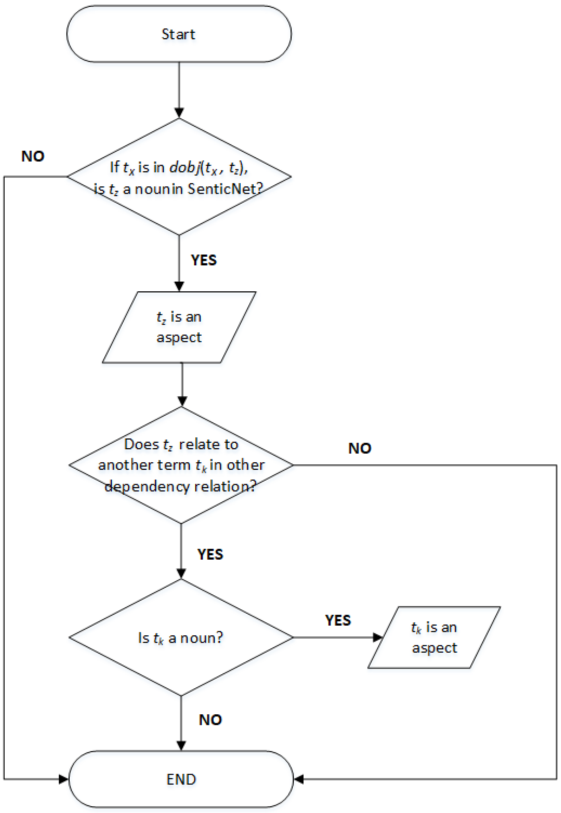


Figure B.4: Rule 3 and 4

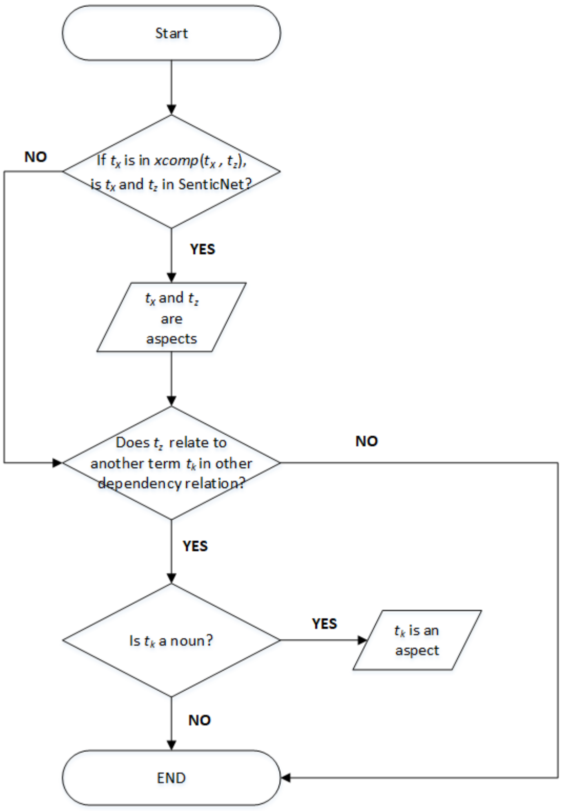


Figure B.5: Rule 5

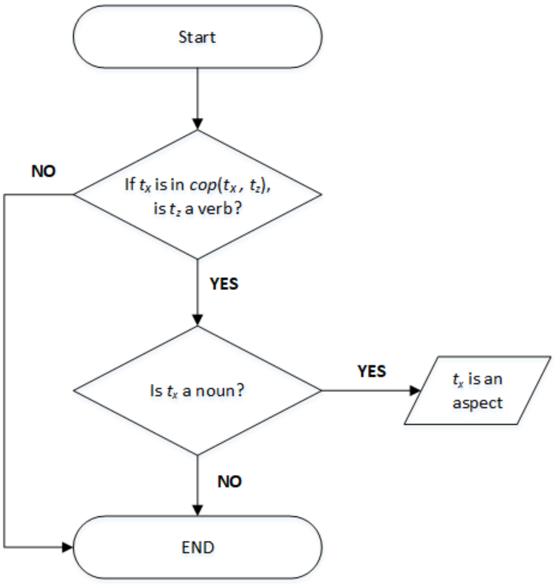


Figure B.6: Rule 6

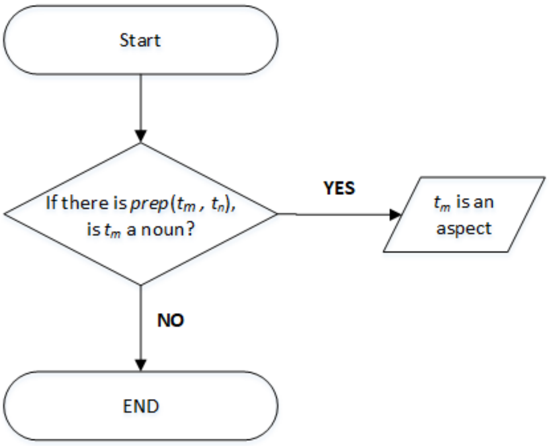


Figure B.7: Rule 7

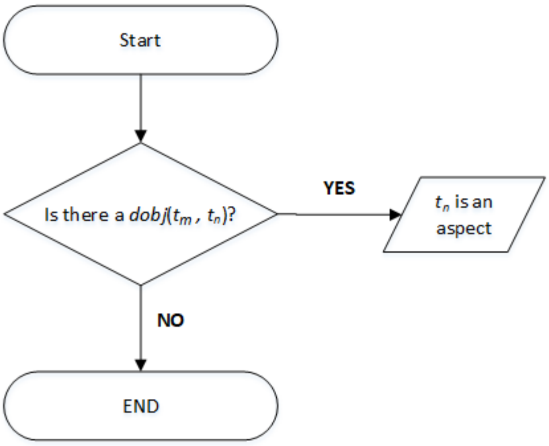


Figure B.8: Rule 8

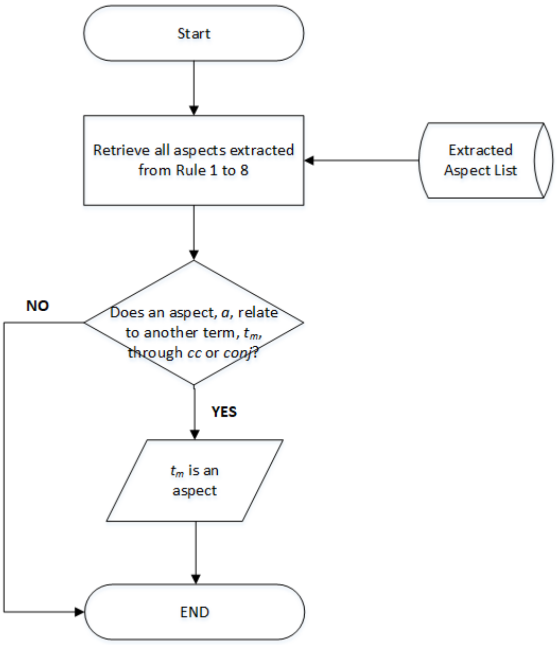

Start
Retrieve all aspects extracted from Rule 1 to 8
Extracted Aspect List
Does an aspect, *a*, relate to another term, t_m, through *cc* or *conj*?
NO
YES
t_m is an aspect
END

Figure B.9: Rule 9

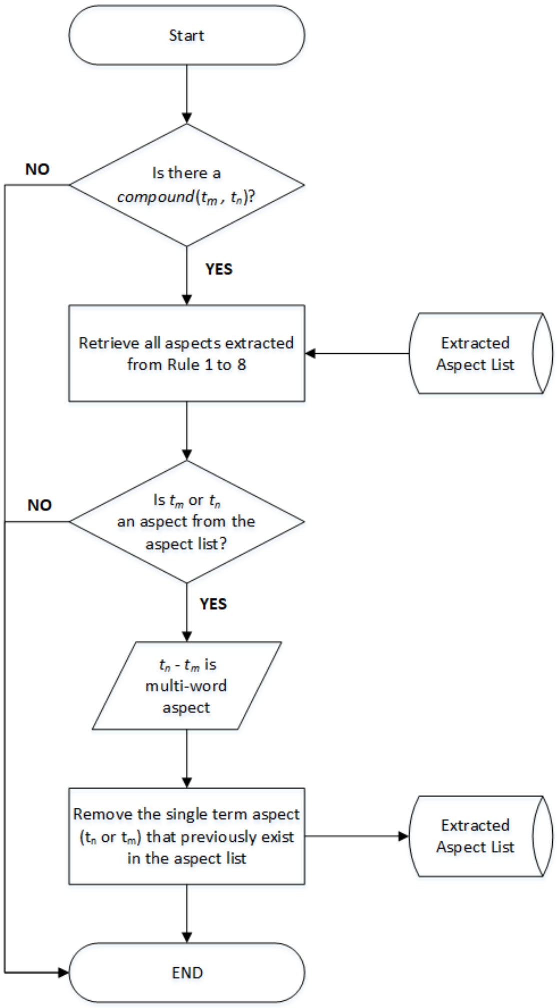

Start
Is there a *compound*(t_m , t_n)?
NO
YES
Retrieve all aspects extracted from Rule 1 to 8
Extracted Aspect List
Is t_m or t_n an aspect from the aspect list?
NO
YES
t_n - t_m is multi-word aspect
Remove the single term aspect (t_n or t_m) that previously exist in the aspect list
Extracted Aspect List
END

Figure B.10: Rule 10

Bibliography

Aciar, S., Zhang, D., Simoff, S., and Debenham, J. (2007). Informed recommender: Basing recommendations on consumer product reviews. *Intelligent Systems, IEEE*, 22(3):39–47.

Bancken, W., Alfarone, D., and Davis, J. (2014). Automatically detecting and rating product aspects from textual customer reviews. In *Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing at ECML/PKDD*, pages 1–16.

Beilin, L. and Yi, S. (2013). Survey of personalized recommendation based on society networks analysis. In *Information Management, Innovation Management and Industrial Engineering (ICIII), 2013 6th International Conference on*, volume 3, pages 337–340. IEEE.

Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. In *Icml*, volume 98, pages 46–54.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Bridge, D., Göker, M. H., McGinty, L., and Smyth, B. (2005). Case-based recommender systems. *The Knowledge Engineering Review*, 20(3):315–320.

Cai, J., Luo, J., Wang, S., and Yang, S. (2018). Feature selection in machine learning: A new perspective. *Neurocomputing*, 300:70–79.

Cambria, E., Livingstone, A., and Hussain, A. (2012). The hourglass of emotions. In *Cognitive behavioural systems*, pages 144–157. Springer.

Cambria, E., Olsher, D., and Rajagopal, D. (2014). Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis. In *Twenty-eighth AAAI conference on artificial intelligence*.

Cao, B., Shen, D., Wang, K., and Yang, Q. (2010). Clickthrough log analysis by collaborative ranking. In *AAAI*.

Celma, Ò. and Cano, P. (2008). From hits to niches?: or how popular artists can bias music recommendation and discovery. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, page 5. ACM.

Chen, G. and Chen, L. (2014). Recommendation based on contextual opinions. In *User Modeling, Adaptation, and Personalization*, pages 61–73. Springer.

Chen, L. and Wang, F. (2013). Preference-based clustering reviews for augmenting e-commerce recommendation. *Knowledge-Based Systems*, 50:44–59.

Chen, Y. Y., Ferrer, X., Wiratunga, N., and Plaza, E. (2014). Sentiment and preference guided social recommendation. In *Case-Based Reasoning Research and Development*, pages 79–94. Springer.

Choi, K., Yoo, D., Kim, G., and Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, 11(4):309–317.

Christopher, D. M., Prabhakar, R., and Hinrich, S. (2008). Introduction to information retrieval. *An Introduction To Information Retrieval*, 151:177.

Cremonesi, P., Koren, Y., and Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 39–46. ACM.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Ding, Y. (2011). Applying weighted pagerank to author citation networks. *Journal of the American Society for Information Science and Technology*, 62(2):236–245.

Dixon, P. M., Weiner, J., Mitchell-Olds, T., and Woodley, R. (1987). Bootstrapping the gini coefficient of inequality. *Ecology*, 68(5):1548–1551.

Dong, R., O'Mahony, M. P., Schaal, M., McCarthy, K., and Smyth, B. (2016). Combining similarity and sentiment in opinion mining for product recommendation. *Journal of Intelligent Information Systems*, 46(2):285–312.

Dong, R., O'Mahony, M. P., and Smyth, B. (2014). Further experiments in opinionated product recommendation. In *Case-Based Reasoning Research and Development*, pages 110–124. Springer.

Dong, R., Schaal, M., O'Mahony, M., McCarthy, K., and Smyth, B. (2013). Opinionated product recommendation. In *Inter. Conf. on Case-Based Reasoning.*

Drummond, G. and Ensor, J. (2006). *Introduction to marketing concepts*. Routledge.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American statistical association*, 56(293):52–64.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational linguistics*, 19(1):61–74.

Eirinaki, M., Pisal, S., and Singh, J. (2012). Feature-based opinion mining and ranking. *Journal of Computer and System Sciences*, 78(4):1175–1184.

Esparza, S. G., O'Mahony, M. P., and Smyth, B. (2011). Effective product recommendation using the real-time web. In *Research and Development in Intelligent Systems XXVII*, pages 5–18. Springer.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proc. Language Resources and Evaluation Conference*, pages 417–422.

Ferrer, X., Chen, Y. Y., Wiratunga, N., and Plaza, E. (2014). Preference and sentiment guided social recommendations with temporal dynamics. In *Research and Development in Intelligent Systems XXXI*, pages 101–116. Springer.

Fisher, R. A. (1956). Statistical methods and scientific inference.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92.

Ganapathibhotla, M. and Liu, B. (2008). Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 241–248. Association for Computational Linguistics.

Ganu, G., Kakodkar, Y., and Marian, A. (2013). Improving the quality of predictions using textual information in online user reviews. *Information Systems*, 38(1):1–15.

Gao, M. and Cui, B. (2016). Literature review on product distinctiveness evaluation and consumer choice based on need for uniqueness. *American Journal of Industrial and Business Management*, 6(07):840.

Gori, M., Pucci, A., Roma, V., and Siena, I. (2007). Itemrank: A random-walk based scoring algorithm for recommender engines. In *IJCAI*, volume 7, pages 2766–2771.

Guy, I. (2015). Social recommender systems. In *Recommender Systems Handbook*, pages 511–543. Springer.

Haruna, K., Akmar Ismail, M., Suhendroyono, S., Damiasih, D., Pierewan, A., Chiroma, H., and Herawan, T. (2017). Context-aware recommender system: A review of recent developmental process and future research direction. *Applied Sciences*, 7(12):1211.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 22(1):5–53.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine learning*, 11(1):63–90.

Horsburgh, B., Craw, S., Massie, S., and Boswell, R. (2011). Finding the hidden gems: Recommending untagged music. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2256.

Htay, S. S. and Lynn, K. T. (2013). Extracting product features and opinion words using pattern knowledge in customer reviews. *The Scientific World Journal*, 2013.

Hu, M. and Liu, B. (2004). Mining and summarising customer reviews. In *Proc. of ACM SIGKDD Inter. Conf. on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177.

Jakob, N. and Gurevych, I. (2010). Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1035–1045. Association for Computational Linguistics.

Jamroonsilp, S. and Prompoon, N. (2013). Analyzing software reviews for software quality-based ranking. In *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2013 10th International Conference on*, pages 1–6. IEEE.

Jannach, D., Lerche, L., and Gdaniec, M. (2013). Re-ranking recommendations based on predicted short-term interests–a protocol and first experiment. In *ITWP 2013: Proceedings of the workshop Intelligent Techniques for Web Personalization and Recommender Systems at AAAI 2013*. Citeseer.

Jannach, D., Zanker, M., Felfernig, A., and Friedrich, G. (2010). *Recommender systems: an introduction*. Cambridge University Press.

Japkowicz, N. and Shah, M. (2011). *Evaluating learning algorithms: a classification perspective*. Cambridge University Press.

Jawaheer, G., Weller, P., and Kostkova, P. (2014). Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 4(2):8.

Jin, W., Ho, H. H., and Srihari, R. K. (2009). A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 465–472. Citeseer.

Jindal, N. and Liu, B. (2008). Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230. ACM.

Justeson, J. S. and Katz, S. M. (1995). Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural language engineering*, 1(01):9–27.

Kaminskas, M. and Bridge, D. (2017). Diversity, serendipity, novelty, and coverage: A survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1):2.

Kang, Y. and Zhou, L. (2017). Rube: Rule-based methods for extracting product features from online consumer reviews. *Information & Management*, 54(2):166–176.

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 4(1):1.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, (8):30–37.

Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Lenat, D. B. and Guha, R. V. (1989). Building large knowledge-based systems; representation and inference in the cyc project.

Li, F., Han, C., Huang, M., Zhu, X., Xia, Y.-J., Zhang, S., and Yu, H. (2010). Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661. Association for Computational Linguistics.

Li, G. and Chen, Q. (2016). Exploiting explicit and implicit feedback for personalized ranking. *Mathematical Problems in Engineering*, 2016.

Li, S., Zha, Z.-J., Ming, Z., Wang, M., Chua, T.-S., Guo, J., and Xu, W. (2011). Product comparison using comparative relations. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1151–1152. ACM.

Li, Z., Zhang, M., Ma, S., Zhou, B., and Sun, Y. (2009). Automatic extraction for product feature words from comments on the web. In *Information Retrieval Technology*, pages 112–123. Springer.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Liu, B. (2015). *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge University Press.

Liu, H., He, J., Wang, T., Song, W., and Du, X. (2013). Combining user preferences and user opinions for accurate recommendation. *Electronic Commerce Research and Applications*, 12(1):14–23.

Liu, Q., Gao, Z., Liu, B., and Zhang, Y. (2015). Automated rule selection for aspect extraction in opinion mining. In *International Joint Conference on Artificial Intelligence (IJCAI).*

Lops, P., De Gemmis, M., and Semeraro, G. (2011). Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*, pages 73–105. Springer.

Lorenzi, F., Ricci, F., Tostes, R., and Brasil, R. (2005). Case-based recommender systems: A unifying view. *Lecture notes in computer science*, 3169:89.

Mitchell, T. (1997). Machine learning, mcgraw-hill higher education. *New York*.

Miyahara, K. and Pazzani, M. J. (2000). Collaborative filtering with the simple bayesian classifier. In *Pacific Rim International conference on artificial intelligence*, pages 679–689. Springer.

Moghaddam, S. and Ester, M. (2010). Opinion digger: An unsupervised opinion miner from unstructured product reviews. In *Proc. Inter. Conf. on Information and Knowledge Management*, CIKM '10.

Moghaddam, S. and Ester, M. (2012). On the design of lda models for aspect-based opinion mining. In *Proc. Inter. Conf. on Information and Knowledge Management*, CIKM '12.

Moling, O., Baltrunas, L., and Ricci, F. (2012). Optimal radio channel recommendations with explicit and implicit feedback. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 75–82. ACM.

Muhammad, A. (2016). Contextual lexicon-based sentiment analysis for social media.

Muhammad, A., Wiratunga, N., and Lothian, R. (2016). Contextual sentiment analysis for social media genres. *Knowledge-Based Systems*.

Muhammad, K., Lawlor, A., Rafter, R., and Smyth, B. (2015). Great explanations: Opinionated explanations for recommendations. In *Case-Based Reasoning Research and Development*, pages 244–258. Springer.

Nakagawa, H. and Mori, T. (2002). A simple but powerful automatic term extraction method. In *COLING-02 on COMPUTERM 2002: Second International Workshop on Computational Terminology - Volume 14*, COMPUTERM '02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nemenyi, P. (1962). Distribution-free multiple comparisons. In *Biometrics*, volume 18, page 263. INTERNATIONAL BIOMETRIC SOC 1441 I ST, NW, SUITE 700, WASHINGTON, DC 20005-2210.

Nowlis, S. M. and Simonson, I. (1996). The effect of new product features on brand choice. *Journal of marketing research*, pages 36–46.

Owoputi, O., O'Connor, B., Dyer, C., Gimpel, K., Schneider, N., and Smith, N. A. (2013). Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Pacula, M. (2009). A matrix factorization algorithm for music recommendation using implicit user feedback.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: bringing order to the web.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Parra, D., Karatzoglou, A., Amatriain, X., and Yavuz, I. (2011). Implicit feedback recommendation via implicit-to-explicit ordinal logistic regression mapping. *Proceedings of the CARS-2011*.

Pazzani, M. J. and Billsus, D. (2007). Content-based recommendation systems. In *The adaptive web*, pages 325–341. Springer.

Popescu, A. and Etzioni, O. (2007). Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28.

Popowich, F. (2005). Using text mining and natural language processing for health care claims processing. *ACM SIGKDD Explorations Newsletter*, 7(1):59–66.

Poria, S., Cambria, E., and Gelbukh, A. (2016). Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Poria, S., Cambria, E., Ku, L.-W., Gui, C., and Gelbukh, A. (2014). A rule-based approach to aspect extraction from product reviews. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, pages 28–37.

Potisuk, S. (2010). Typed dependency relations for syntactic analysis of thai sentences. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation*.

Qiu, G., Liu, B., Bu, J., and Chen, C. (2011). Opinion word expansion and target extraction through double propagation. *Computational linguistics*, 37(1):9–27.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Rana, T. A. and Cheah, Y.-N. (2017). A two-fold rule-based model for aspect extraction. *Expert Systems with Applications*.

Ricci, F., Rokach, L., and Shapira, B. (2011). *Introduction to recommender systems handbook*. Springer.

Ricci, F., Rokach, L., and Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender systems handbook*, pages 1–34. Springer.

Ronen, R., Koenigstein, N., Ziklik, E., and Nice, N. (2013). Selecting content-based features for collaborative filtering recommenders. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 407–410. ACM.

Salganik, M. J., Dodds, P. S., and Watts, D. J. (2006). Experimental study of inequality and unpredictability in an artificial cultural market. *science*, 311(5762):854–856.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM.

Schouten, K. and Frasincar, F. (2016). Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge & Data Engineering*, (1):1–1.

Schröder, G., Thiele, M., and Lehner, W. (2011). Setting goals and choosing metrics for recommender system evaluations. In *In CEUR Workshop Proc.*, volume 811, pages 78–85.

Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer.

Smyth, B. (2007). Case-based recommendation. In *The adaptive web*, pages 342–376. Springer.

Speer, R. and Havasi, C. (2012). Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.

Su, Q., Xu, X., Guo, H., Guo, Z., Wu, X., Zhang, X., Swen, B., and Su, Z. (2008). Hidden sentiment association in chinese web opinion mining. In *Proceedings of the 17th international conference on World Wide Web*, pages 959–968. ACM.

Su, X. and Khoshgoftaar, T. M. (2009). A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009:4.

Thelwall, M., Buckley, K., and Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology*, 63(1):163–173.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., and Kappas, A. (2010). Sentiment strength detection in short informal text. *Journal of the Association for Information Science and Technology*, 61(12):2544–2558.

Tsur, O. and Rappoport, A. (2009). Revrank: A fully unsupervised algorithm for selecting the most helpful book reviews. In *ICWSM*.

Tubishat, M., Idris, N., and Abushariah, M. A. (2018). Implicit aspect extraction in sentiment analysis: Review, taxonomy, oppportunities, and open challenges. *Information Processing & Management*, 54(4):545–563.

Tukey, J. W. (1949). Comparing individual means in the analysis of variance. *Biometrics*, pages 99–114.

Turney, P. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proc. Annual Meeting on Association for Computational Linguistics*, pages 417–424.

Vargas-Govea, B., González-Serna, G., and Ponce-Medellın, R. (2011). Effects of relevant contextual features in the performance of a restaurant recommender system. *ACM RecSys*, 11.

Vasudevan, S. and Chakraborti, S. (2014). Mining user trails in critiquing based recommenders. In *Proc. Inter. Conf. on World Wide Web Companion*, pages 777–780.

Wang, F. and Chen, L. (2012). Recommendation based on mining product reviewers preference similarity network. In *Proceedings of 6th SNAKDD Workshop, Beijing.*

Wang, F., Pan, W., and Chen, L. (2013). Recommendation for new users with partial preferences by integrating product reviews with static specifications. In *User Modeling, Adaptation, and Personalization*, pages 281–288. Springer.

Wang, H. and Wang, W. (2014). Product weakness finder: an opinion-aware system through sentiment analysis. *Industrial Management & Data Systems*, 114(8):1301–1320.

Wang, W., Pan, S. J., Dahlmeier, D., and Xiao, X. (2017). Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322.

Wang, W., Wang, H., and Song, Y. (2016). Ranking product aspects through sentiment analysis of online reviews. *Journal of Experimental & Theoretical Artificial Intelligence*, pages 1–20.

Weber, R. O., Ashley, K. D., and Brüninghaus, S. (2005). Textual case-based reasoning. *The Knowledge Engineering Review*, 20(3):255–260.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6):80–83.

Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

Wiratunga, N., Koychev, I., and Massie, S. (2004). Feature selection and generalisation for retrieval of textual cases. In *Advances in Case-Based Reasoning*, pages 806–820. Springer.

Xie, S., Wang, G., Lin, S., and Yu, P. S. (2012). Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 823–831. ACM.

Xu, L., Liu, K., Lai, S., Chen, Y., Zhao, J., et al. (2013). Mining opinion words and opinion targets in a two-stage framework. In *ACL (1)*, pages 1764–1773.

Yang, D., Chen, T., Zhang, W., Lu, Q., and Yu, Y. (2012a). Local implicit feedback mining for music recommendation. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 91–98. ACM.

Yang, X., Steck, H., Guo, Y., and Liu, Y. (2012b). On top-k recommendation using social networks. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 67–74. ACM.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420.

Yates, A., Joseph, J., Popescu, A.-M., Cohn, A. D., and Sillick, N. (2008). Shopsmart: product recommendations through technical specifications and user reviews. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1501–1502. ACM.

Yu, J., Zha, Z.-J., Wang, M., and Chua, T.-S. (2011). Aspect ranking: identifying important product aspects from online consumer reviews. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1496–1505. Association for Computational Linguistics.

Yujie, Z. and Licai, W. (2010). Some challenges for context-aware recommender systems. In *Computer Science and Education (ICCSE), 2010 5th International Conference on*, pages 362–365. IEEE.

Zha, Z.-J., Yu, J., Tang, J., Wang, M., and Chua, T.-S. (2014). Product aspect ranking and its applications. *Knowledge and Data Engineering, IEEE Transactions on*, 26(5):1211–1224.

Zhang, C., Wang, K., Lim, E.-p., Xu, Q., Sun, J., and Yu, H. (2015). Are features equally representative? a feature-centric recommendation. In *AAAI*, pages 389–395.

Zhang, K., Cheng, Y., Liao, W.-k., and Choudhary, A. (2012). Mining millions of reviews: A technique to rank products based on importance of reviews. In *Proceedings of the 13th International Conference on Electronic Commerce*, ICEC '11, pages 12:1–12:8, New York, NY, USA. ACM.

Zhang, K., Narayanan, R., and Choudhary, A. N. (2010a). Voice of the customers: Mining online customer reviews for product feature-based ranking. *WOSN*, 10:11–11.

Zhang, L., Liu, B., Lim, S. H., and O'Brien-Strain, E. (2010b). Extracting and ranking product features in opinion documents. In *Proceedings of the 23rd international conference on computational linguistics: Posters*, pages 1462–1470. Association for Computational Linguistics.

Zhang, Y. and Pennacchiotti, M. (2013). Recommending branded products from social media. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 77–84. ACM.

Zhao, S., Du, N., Nauerz, A., Zhang, X., Yuan, Q., and Fu, R. (2008). Improved recommendation based on collaborative tagging behaviors. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 413–416. ACM.

Zhao, W. X., Li, S., He, Y., Chang, E., Wen, J.-R., and Li, X. Connecting social media to e-commerce: Cold-start product recommendation on microblogs.

Zhao, X. W., Guo, Y., He, Y., Jiang, H., Wu, Y., and Li, X. (2014). We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1935–1944. ACM.

Zhu, H., Huberman, B., and Luon, Y. (2012). To switch or not to switch: understanding social influence in online choices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2257–2266. ACM.

Zhu, J., Wang, H., Tsou, B. K., and Zhu, M. (2009). Multi-aspect opinion polling from textual reviews. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1799–1802. ACM.

Zhuang, L., Jing, F., and Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM.