Computing Multi-Dimensional Trust by mining E-Commerce Feedback Comments

A thesis submitted in fulfilment of the requirements for the degree of Master of Engineering

Lishan Cui
M.Eng.

School of Computer Science and Information Technology
RMIT University

January 2014

Declaration

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; and, any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Lishan Cui
School of Computer Science and Information Technology
RMIT University
January 2014

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this research thesis. I would like to thank my supervisors, Dr.Xiuzhen Zhang and Professor Yan Wang, for their support, not only with my research and thesis, but also in other events in my life. They have been great supervisors in motivating me on doing high quality research and keeping me on track on progress.

I would like to thank several of my fellow research students, especially Sabrina, Husna and Shafiza. It was of great help to have such kind friends to support me and to socialise with throughout my research study.

The RMIT Research Writing group was also beneficial in my research, allowing me to engage in frequent discussions of my work, which helped to improve the ideas and presentation found in this thesis.

Last but not least, I would like to thank my family and friends for their love, motivation and guidance throughout my entire academic career.

Credits

Portions of the material in this thesis have previously appeared in the following publications:

- X. Zhang, L. Cui, and Y. Wang. Computing Multi-Dimensional Trust by Mining E-Commerce Feedback Comments. *IEEE Transactions on Knowledge and Data Engineering*, Accepted on 5 Nov. 2013. To appear.
- L. Cui, X. Zhang, and Y. Wang. Mining E-Commerce Feedback Comments for Dimension Rating Profiles. In *Proc. The 9th International Conf. on Advanced Data Mining and Applications, ADMA*, 2013.

The thesis was written in the `TeXstudio` editor on Windows 7, and typeset using the LaTeX 2_ε document preparation system. Experimental work was performed using Sun Java. Results were analysed using RStudio, and graphs generated using Microsoft Visio and ggplot2 Lib.

Note

Unless otherwise stated, all fractional results have been rounded to the displayed number of decimal figures.

Contents

List of Figures

List of Tables

Abstract

Reputation-based trust models are widely used in e-commerce applications, and feedback ratings are aggregated to compute sellers' reputation trust scores. The "all good reputation" problem however is prevalent in current reputation systems – reputation scores are universally high for sellers and it is difficult for potential buyers to select trustworthy sellers.

In this thesis, based on the observation that buyers often express opinions openly in free text feedback comments, we have proposed CommTrust, a multi-dimensional trust evaluation model, for computing comprehensive trust profiles for sellers in e-commerce applications. Different from existing multi-dimensional trust models, we compute dimension trust scores and dimension weights automatically via extracting dimension ratings from feedback comments.

Based on the dependency relation parsing technique, we have proposed Lexical-LDA (Lexical Topic Modelling based approach) and DR-mining (Lexical Knowledge based approach) approaches to mine feedback comments for dimension rating profiles. Both approaches achieve significantly higher accuracy for extracting dimension ratings from feedback comments than a commonly used opinion mining approach. Extensive experiments on eBay and Amazon data demonstrate that CommTrust can effectively address the "all good reputation" issue and rank sellers effectively. To the best of our knowledge, our research demonstrates the novel application of combining natural language processing with opinion mining and summarisation techniques in trust evaluation for e-commerce applications.

Chapter 1

Introduction

1.1 Background

There has been a tremendous growth in e-commerce applications, where buyers and sellers conduct transactions on the web. Users are attracted to online-shopping not only due to the convenience in accessing the information of items on-sold, but also because of the availability of other buyers feedback on their purchasing experience, item-related and/or seller-related. All major online-shopping websites encourage buyers to provide feedback, often in the form of ratings along with some textual comments, to facilitate potential transactions.

Reputation reporting systems [Resnick et al., 2000; Xiong and Liu, 2004; Zacharia and Maes, 2000] have been implemented in e-commerce systems such as eBay and Amazon (for third-party sellers), where overall reputation trust scores for sellers are computed by aggregating feedback ratings. In e-commerce environments, reputation mechanisms are related to the ratings that a seller received from buyers. The ratings indicate the ability of the seller to provide satisfactory transactions in the future, which is beneficial to new buyers. For example on eBay, the reputation score for a seller is computed by aggregating buyer feedback ratings in the past 12 months, such as either the total number of positive ratings minus the total number of negative ratings or the percentage of positive ratings out of the total number of positive ratings and negative ratings.[1]

A well-reported issue with the eBay reputation management system is the "all good sellers" problem [Resnick et al., 2000; Resnick and Zeckhauser, 2002] where feedback ratings are over 99% positive on average [Resnick et al., 2000]. Such strong positive bias can hardly guide buyers to select sellers to transact with them. At eBay detailed seller ratings for

[1] http://pages.ebay.com/help/feedback/allaboutfeedback.html

sellers (DSRs) on four aspects of transactions, namely *item as described, communication, postage time*, and *postage and handling charges* are also reported. DSRs are aggregated rating scores on a 1- to 5-star scale. Still the strong positive bias is present – our analysis on sample eBay data shows that on average over 60% of aspect ratings are 4.9 stars. One possible reason for the lack of negative ratings at e-commerce web sites is that users who leave negative feedback ratings can attract retaliatory negative ratings and thus damage their own reputation [Resnick et al., 2000]. Note also that DSRs are not used to compute the overall trust scores for sellers.

The textual comments can provide detailed information that is not available in ratings. Even though buyers leave positive feedback ratings, they still express some disappointment and negativeness in free text feedback comments, often towards specific aspects, or *dimensions* of transactions. For example, a comment like "*The products were as I expected.*" expresses positive opinion towards the Product dimension, whereas the comment "*Delivery was a little slow but otherwise, great service. Recommend highly.*" expresses negative opinion towards the Delivery dimension but a positive rating to the transaction in general.

There are several reasons why comments provide more reliable information. First, ordinal ratings are interpreted differently by different users. Some users tend to rate higher while others tend to rate lower. Secondly, most online shopping websites also allow sellers to rate the buyers to counter-balance the impact of malicious buyers. Since the average rating could affect the sales greatly, sellers may use this mechanism as a weapon to defend their business, rating down buyers who provide low ratings on their purchase. As such, the mechanism effectively leads to pseudo high ratings than what comments are reflecting.

From the buyer's perspective, while the average rating may not be a fully reliable measure, it is the only easily accessible measure. Browsing through tens of pages of comments can be time consuming, and to digest the information is a daunting task, as well. This calls for a better measure to represent the reputation of seller accurately. Such reputation is sometimes referred to as trust, which is defined by Wang and Lim [2008] as "the extent to which one party measures the other party's willingness and ability to act in the measuring party's interest".

By analysing the wealth of information in feedback comments we can uncover buyers' embedded opinions towards different aspects of transactions, and compute comprehensive reputation profiles for sellers. Specifically using the positive and negative subjectivity of opinions towards aspects of transactions as *dimension ratings*, we propose *Comment-based Multi-dimensional trust (CommTrust)*, a fine-grained multi-dimension trust evaluation model

for e-commerce applications.

1.2 Research problems

Different from existing work of computing trust from user ratings, we propose a multi-dimensional trust model based on feedback comments. The trust is decomposed into multiple aspects to represent different dimensions of a transaction, including such as the quality of products or the delivery status of orders. We derive trust dimensions from textual feedback comments and combine customer preferences on each dimension to highlight customer concerns. There are four main research questions:

1. *How can multi-dimensional trust from extracted dimensions and the associated opinion polarity be computed?*

   In e-commerce environments different transactions may have different contexts. The trustworthiness of a seller should be related to forthcoming transactions. How to efficiently compute the trust level for a seller from the sentiment expressed in buyer feedback comments and represent it effectively is our first task.

2. *How can dimensions from online feedback comments that customers have expressed their opinions on be more accurately identified?*

   In e-commerce, sellers provide products and services, and buyers pay for them. During the process of finishing these transactions, the quality of products, communication of sellers (whether the seller has friendly communication with buyers), delivery time (whether the seller delivers items on time) and shipping charges (whether the charges are reasonable) might be some of the dimensions which buyers are interested in. In online feedback comments, different customers describe different aspects of dimensions. How to accurately identify these dimensions expressed in natural language textual comments is our second task.

3. *How can weights for each dimension that extracted from feedback comments be evaluated?*

   In e-commerce, the feedback comments and ratings leaved by buyers are highly noisy. There are many comments are writen from the same buyer and therefore are highly correlated. Some buyers are lenient or harsh raters and therefore their ratings should

be taken with a grain of salt. How to efficiently evaluate the weights of each dimension is our third task.

4. *How can sentiment from textual feedback for each dimension be more accurately classified?*

   Sentiment classification aims to identify view-point from information expressed in text. Whether a piece of text is expressing positive or negative attitude towards associated dimensions of comments need to be identified. How to accurately classify sentiment is our fourth task.

1.3 Thesis objectives and scope

Our work aims to provide a comprehesive trust profiles for sellers that allows buyers to conduct their online shopping based on past experience. Our focus is on extracting dimension ratings from feedback comments and further aggregating these dimension ratings to compute dimension trust scores. The motivation of our research is that online feedback comments contain disdinct informatnion for users to rank sellers, therefore content of comments can be used to reliably evaluate the trustworthiness of sellers.

The contribution of this thesis are:

- We propose to use Comment-based Multi-dimensional trust (CommTrust), a fine-grained multi-dimension evaluation model, to calculate the trust for e-commerce applications. While the model is potentially extensible to target item-specific trust, in this study we focus on computing comprehensive trust profile for sellers.

- We propose an algorithm to identify dimension rating expresses from feedback comments by applying lexicon-based opinion mining techniques [Pang and Lee, 2008] in combination with dependency relation analysis, a tool recently developed in natural language processing (NLP) [De Marneffe et al., 2006; De Marneffe and Manning, 2008].

- We tackle the four research questions by two approaches:

  1. The topic modelling approach is applied to develop the Lexical-LDA algorithm for grouping dimension rating extraction and trust computation. Lexical LDA makes use of two types of lexical knowledge based on dependency relations for clustering dimension expressions into dimensions so as to produce meaningful cluster. The

first lexical knowledge is that the co-occurrence of dimension expressions with respect to a same modifier across comments can provide more meaningful contexts for dimension expressions, compare to add on counts of dimension expressions by comments. The second knowledge is that the dimension expressions extracted from the same comment are very unlikely about the same topic. Based on these two types of lexical knowledge, we revised Latent Dirichlet Allocation (LDA) [Blei et al., 2003] to develop the Lexical-LDA algorithm.

2. With the seed dimension words we propose Dimension Rating mining (DR-mining), a knowledge-based approach that incorporates domain knowledge, meta-data, and general grammatical patterns to accurately identifying dimension rating expressions from feedback comments. The matrix factorisation technique applied to automatically compute trust weights.

To the best of our knowledge, CommTrust is the first piece of work that computes fine-grained multidimensional trust profiles automatically by mining feedback comments.

The rest of this thesis is organized as follows.

In Chapter 2, the necessary background knowledge of the trust evaluation, sementic analysis, and text comments mining related works is introduced.

In Chapter 3, we propose the comment-based multi-dimensional trust (CommTrust) model to identify trustworthy and reliable sellers..

In Chapter 4, we present topic modelling approach to mining feedback comments for dimension rating profiles. We propose the Lexical-LDA algorithm to conduct dimension rating extraction and trust computation.

In Chapter 5, we propose a knowledge-based approach that incorporates domain knowledge, meta-data, and general grammatical patterns to mining feedback comments for dimension rating profiles. We formulate the problem of computing dimension weights from ratings as a factor analytic problem and propose a matrix factorisation technique to automatically compute weights for dimensions from the sparse and noisy dimension rating matrix.

We conclude out study in Chapter 6, where the work of the thesis is summarised, particularly in relation to the research questions. Furth more the future research problems are discussed.

Chapter 2

Literature Review

Related work for our research falls into four main areas: 1) computational approaches to trust, especially reputation-based trust evaluation and recent developments in fine-grained trust evaluation; 2) e-commerce feedback comments analysis and more generally mining opinions on movie reviews, product reviews and other forms of free text documents; 3) aspect opinion extraction and summarisation on movie reviews, product reviews and other forms of free text; and 4) applications of the matrix factorisation technique for recommender systems and other data mining tasks.

2.1 Computational trust evaluation

The strong positive rating bias in the eBay reputation system has been well documented in literature [Resnick et al., 2000; Resnick and Zeckhauser, 2002; ODonovan et al., 2007], although no effective solutions have been reported. Notably in [ODonovan et al., 2007] it is proposed to examine feedback comments to bring seller reputation scores down to a reasonable scale, where comments that do not demonstrate explicit positive ratings are deemed negative ratings on transactions. Ratings on transactions are further aggregated as the overall trust scores for sellers. In this study on the other hand, our focus is on extracting dimension ratings from feedback comments and further aggregating these dimension ratings to compute dimension trust scores.

The notion of computational trust is essential to ensure the operation of open systems. Similar to that buyers and sellers are referred to as individuals in e-commerce applications, terms like peers and agents are often used to refer to individuals in open systems in various applications in the trust evaluation literature. Trust is the subjective probability with which

an individual assesses that another individual performs a given action [Jøsang et al., 2007]. In [Ramchurn et al., 2004] a comprehensive overview of trust models is provided. Individual level trust models are aimed to compute the reliability of peers and assist buyers in their decision making [Yu and Singh, 2002; Schillo, 2000; Sabater and Sierra, 2001] whereas system level models are aimed to regulate the behaviour of peers, prevent fraudsters and ensure system security [Ramchurn et al., 2004]. Reputation is a collective measure of trustworthiness computed from referrals or ratings from members in a community [Jøsang et al., 2007]. Reputation-based trust models are a class of trust models that aim to use public reputation profiles of peers to promote good behaviours and ensure security and reliability of open systems [Resnick et al., 2000; Ramchurn et al., 2004; Yu and Singh, 2002; Schillo, 2000; Jøsang et al., 2007; Kamvar et al., 2003; Rettinger et al., 2011; Wang et al., 2012; Xiong and Liu, 2003; 2004], and have been widely used in e-commerce systems [Resnick et al., 2006], peer-to-peer networks [Xiong and Liu, 2004], and multi-agent systems [Ramchurn et al., 2004; Wang and Singh, 2006].

Accurately computing individual reputation requires effective approaches to gathering and aggregating ratings for individuals. Rating aggregation algorithms include simple positive feedback percentage or average of star ratings as in the eBay and Amazon reputation systems [Resnick et al., 2006], the Beta reputation based on statistical distribution assumption for ratings [Jøsang and Ismail, 2002], as well as more advanced models like Kalman inference [Wang et al., 2012], which also computes trust score variance and confidence level. More sophisticated reputation models consider factors like time, where recent feedback ratings carry more weights [Sabater and Sierra, 2001; Wang and Singh, 2006]. PeerTrust [Xiong and Liu, 2003; 2004] is a framework for peer-to-peer systems where contextual factors are considered for computing trust scores and weights for peers. The trust score for a peer is computed by combining a weighted average satisfaction amount that the peer receives from transactions and a weighted trust measure by community-based characteristics. Contextual factors considered for computing trust scores and weights include the total number of transactions a peer has completed and the credibility of feedback sources. The EigenTrust algorithm [Kamvar et al., 2003] uses a rating matrix representation for local trust scores and computes the global ratings for peers based on finding the principal eigenvector of the rating matrix. All existing models previously discussed assume that feedback ratings are readily available and focus on aggregation algorithms [Sabater and Sierra, 2001; Wang et al., 2012; Xiong and Liu, 2003; 2004; Wang and Singh, 2006]. A couple of studies focus on gathering ratings through social networks [Yu and Singh, 2002; Schillo, 2000]. Nevertheless ratings

are assumed available rather than obtained via data mining.

The multi-dimensional approach to fine-grained trust computation has been studied in the area of agent technologies [Sabater and Sierra, 2001; Griffiths, 2005; Reece et al., 2007], where the overall reputation score for an agent is computed by aggregating dimension reputations. In [Sabater and Sierra, 2001], individual, social and ontological reputations are computed from factors such as delivery date and quality, and their ratings are then combined to form an overall score. In [Griffiths, 2005] the dimension scores are computed from direct experience of individual agents, and then aggregated by weighted summation. Reece et al. [Reece et al., 2007] presented a probabilistic approach considering the correlation among dimension during aggregation. In all these multi-dimensional trust models however, weightings for dimension trust are either not considered or assumed given.

Other approaches to fine-grained trust computation have also been proposed in literature [Rettinger et al., 2011; Wang and Lim, 2008; Zhang and Fang, 2007; Zhang et al., 2012a;b], where specific factors for individual and transaction contexts are considered. However, many factors considered in these models such as attributes associated with products and attributes regarding the interactions between sellers and products, are not readily available in e-commerce applications.

In CommTrust we focus on mining dimension ratings from free text comments, and furthermore our trust evaluation model computes fine-grained dimension trust scores and dimension weights, both from the dimension rating matrix obtained by mining feedback comments.

2.2 Feedback comment analysis

The success of e-commerce applications, such as eBay and Amazon, depends highly on the availability of user interaction. Usually, reputable sellers attract a large user population to transact with them and leave comments afterwards. Intuitively, one can use a reputation score to quantify how good a seller is at providing good services. However, the strong positive rating bias in reputation system has been noted in literature [Resnick et al., 2000; Resnick and Zeckhauser, 2002; ODonovan et al., 2007]. There have been studies on analysing feedback comments in e-commerce applications to capture negativity information to provide reasonable reputation range score for sellers [ODonovan et al., 2007; Gamon, 2004; Hijikata et al., 2007; Lu et al., 2009].

ODonovan et al. [2007] and Gamon [2004] focus on sentiment classification of feedback comments. It is demonstrated that feedback comments are noisy and therefore analysing them is a challenging problem. ODonovan et al. [2007] tackled the problem of excessive positive bias of feedback ratings in eBay by extracting more negative feedback from free text comments. Comments do not demonstrate explicit positive ratings or missing aspect comments are deemed negative ratings on transactions. Ratings on transactions are further aggregated as the overall trust scores for sellers. Gamon [2004] formulated sentiment classification of customer feedback comments as a special case of text categorisation. As a result, more feedback ratings are classified to be negative and the feedback ratings on eBay are brought to a more reasonable scale.

In [Hijikata et al., 2007], a technique for summarising feedback comments is presented, aiming to filter out courteous comments that do not provide real feedback. They noticed, feedback comments include not only comments presenting real opinions but also many stereotyped sentences, clauses or phrases such as expressions for thanks or expressions of courtesy, hereinafter all of which are referred as "descriptions of courtesy". They propose a social summarization method (SS method) for summarizing feedback comments. This method uses social relationships in an online auction for summarizing one sellers feedback comments. This method does not focus on the seller but focuses on a buyer who bought an item from the seller. This method compares the feedback comment on the target seller written by a certain buyer to the feedback comments on the sellers other than the target seller written by the buyer. Then it produces a summary by extracting two types of descriptions. One is a description that appears only in the feedback comment on the target seller and the other is a description that appears in the feedback comments on the sellers other than the target seller but does not appear in the feedback comment on the target seller. By this method, the descriptions of courtesy can be eliminated without deleting descriptions which are seemed that the buyers wrote from their real feelings.

Lu et al. [2009] focuses on generating "rated aspect summary" from eBay feedback comments, aiming to discover different perspectives towards their aggregated ratings. This kind of decomposition is quite useful because different users may have quite different needs and the overall ratings are generally not informative enough. A statistical technique named structured Probabilistic Latent Semantic Analysis (PLSA) was proposed to discover aspects and their ratings from user comments. However their statistical generative model is based regression on the overall transaction ratings and the resultant aspect ratings are likely biased due to the positive bias in transaction ratings.

With CommTrust rather than simply classifying comments into positive or negative as in [ODonovan et al., 2007], we mine the text comments to extract dimensions and their associated feedback orientations hidden in the free text, which is free from the positive bias in the overall transaction ratings. Different from [Lu et al., 2009], which adopted a statistical generative learning model, CommTrust adopts a more knowledge-based approach making use of the deeper lexical knowledge of dependency relation from the Stanford natural language parser [De Marneffe et al., 2006; Kubler et al., 2009] between candidate dimensional words and opinion words. More importantly our work aims at inferring both the dimension ratings and dimension weights rather than generating an aggregated summary inferred from overall ratings. Our approach is complementary to the statistical approach and potentially can greatly improve the computation efficiency and effectiveness of statistical models.

2.3 Aspect opinion extraction and summarisation

More generally our work is related to opinion mining and sentiment analysis on free text documents, especially opinion mining in product reviews and movie reviews. In these studies, product or movie features and the opinions towards them are extracted. Summaries are produced by selecting and re-organising sentences according to the extracted features.

Review mining and summarization is the task of producing a sentiment summary, which consists of sentences from reviews that capture the authors opinion. Review summarization is interested in features or aspects on which customers have opinions. It also determines whether the opinions are positive or negative. This makes it differ from traditional text summarization. A comprehensive overview of the field is presented in [Pang and Lee, 2008; Liu, 2012]. Most existing works on review mining and summarization mainly focus on product reviews. For example, [Hu and Liu, 2004b; Popescu and Etzioni, 2005; Shi and Chang, 2006] concentrated on mining and summarizing reviews by extracting opinion sentences regarding product aspects.

Hu and Liu [2004a;b] aim to summarize all the customer reviews of a product to help a potential customer make a decision on whether to buy the product. They proposed to extract nouns and noun phrases as candidate aspects and then apply association rule mining techniques to compute the aspects and their associated opinions. The NLProcessor linguistic parser (NLProcessor 2000) is applied to parse each sentence and identify simple nouns/noun phrases as product aspects by yielding the part-of-speech tag [Manning and Schütze, 1999] of each word. Association rule mining [Agrawal and Srikant, 1994] is used to find all frequent

itemsets (a set of words or a phrase that occurs together). They extract opinion words as nearby *adjective*, which adjacent adjective that modifies the noun/noun phrases that is a frequent feature. WordNet [Stark and Riesenfeld, 1998; Miller et al., 1990] is used to identify opinion orientation for all the adjectives in the opinion word list. For those adjectives that WordNet cannot recognize, they are discarded as they may not be valid words. For the case that the synonyms or antonyms of an adjective have different known semantic orientations, use the first found orientation as the orientation for the given adjective. They also checked negation words, which appearing close around the opinion word, to opposite the opinion orientation of the sentence. The *close* means that the word distance between a negation word and the opinion word should not exceed a threshold(they set it to 5).

Carenini et al. [2005] proposed feature extraction for capturing knowledge from product reviews. Their method is based on the results of Hu and Liu [2004b], then mapped to the user-defined taxonomy features hierarchy thereby eliminating redundancy and providing conceptual organization.

Popescu and Etzioni [2005] developed an unsupervised information extraction system called OPINE, which extracted product aspects and opinions from reviews. Similar to previous works, OPINE extracts noun phrases from reviews and retains those with frequency greater than an experimentally set threshold, and then OPINEs feature assessor assesses those noun phrases for extracting explicit aspects. The assessor evaluates a noun phrase by computing a Point-wise Mutual Information score between the phrase and meronymy discriminators associated with the product class.

Somprasertsri and Lalitrojwong [2010] proposed an approach to extract product aspects and to identify the opinions associated with these aspects from reviews through syntactic information based on dependency analysis. Similar to Hu and Liu [2004a;b], they extract noun or noun phrase as a potential product aspect, adjective as a potential opinion word. The aspect-opinion pair candidates are identify by dependency path, not using the distance between adjective and noun/noun phrase.

In another domain, Zhuang et al. [2006] proposed a multi-knowledge based approach, which integrates WordNet, statistical analysis and movie knowledge, for movie review mining and summarization. They used WordNet, movie casts and labeled training data to generate a keyword list for finding aspects and opinions. The grammatical rules between aspect words and opinion words were applied to identify the valid aspect-opinion pairs.

To identify the expressions of opinions associated with aspects, some researchers considered that a product aspect and its opinion words usually co-occur within a certain distance

in the sentence. Hu and Liu [2004b] focused on adjacent adjectives that modify aspect nouns or noun phrases. They use adjacent adjectives as opinion words that associated with aspects. Kim and Hovy [2004] explored the following four sizes of regions which may contain both of product aspects and their opinions. The four regions are: (1) full sentences; (2) words between the opinion holder and the topic; (3) region 2 +/- two words; and (4) from the first word behind the holder to the end of sentences. In other research, Popescu and Etzioni [2005] apply manual extraction rules in order to find the opinion words. This idea is similar to that of [Hu and Liu, 2004b] and [Kim and Hovy, 2004], but instead of using a window of size or adjacent adjectives they define extraction rules to find the expressions of opinions. More recently, Zhuang et al. [2006], Qiu et al. [2009], Qiu et al. [2011] and Somprasertsri and Lalitrojwong [2010] apply dependency relation analysis to extract aspect opinions. However these work do not group aspect opinion expressions into clusters.

Some work groups aspects into clusters, assuming aspect opinion expressions are given [Zhai et al., 2011]. Recently a semi-supervised algorithm [Mukherjee and Liu, 2012] was proposed to extract aspects and group them into meaningful clusters as supervised by user input seed words. Unsupervised topic modelling-based techniques have been developed to jointly model opinions and aspects (or topics), based on either the probabilistic Latent Semantic Analysis (pLSA) [Hofmann, 1999] or Latent Dirichlet Allocation (LDA) [Blei et al., 2003]. The models differ in granularities [Mei et al., 2007; Titov and McDonald, 2008b; Lin and He, 2009; Titov and McDonald, 2008a; Brody and Elhadad, 2010] and how aspects and opinions interact [Lin and He, 2009; Brody and Elhadad, 2010; Mei et al., 2007; Zhao et al., 2010]. All these existing work however are based on the unigram representation of documents and none of them make use of any lexical knowledge.

There has been some recent work on computing aspect ratings from overall ratings in e-commerce feedback comments or reviews [Lu et al., 2009; Wang et al., 2010; 2011]. Their aspect ratings and weights are computed based on regression from overall ratings and the positive bias in overall ratings is not the focus.

For word sentiment classification, the basic approach is to assemble a small amount of seed words by hand, sorted by polarity into two lists - positive and negative - and then to grow this by adding words obtained from WordNet [Miller et al., 1990]. This approach assumes, synonyms of positive words are mostly positive and antonyms mostly negative. Antonyms of negative words are added to the positive list, and synonyms to the negative one. Opinion lexicons like SentiWordNet have been developed in linguistic studies, and have been widely used for various sentiment analysis tasks. SentiWordNet [Esuli and Sebastiani, 2006] is

an opinion lexicon derived from the WordNet database where each term is associated with numerical scores indicating positive and negative sentiment information. Ohana and Tierney [2009] states the SentiWordNet lexical resource could be used as an important resource for the problem of automatic sentiment classification tasks.

In CommTrust we develop Lexical-LDA and DR-mining approaches make use of the lexical knowledge of dependency relation analysis to identify dimension ratings in feedback comments. SentiWordNet is applied to determine sentiment orientation.

2.4 Matrix factorisation

Matrix factorisation is a general factor analytic technique, and has been widely applied in various areas, especially in information retrieval [Deerwester et al., 1990] and recommender systems [Hofmann, 2004; Koren et al., 2009; Koren, 2010]. In information retrieval, latent Semantic Indexing (LSI)[Deerwester et al., 1990] is a method for automatic indexing and retrieval, where singular value decomposition (SVD) is applied to decompose a term by document matrix into a set of orthogonal factors from which the original matrix can be approximated by linear combination. Similarity between documents, queries, and terms can be easily represented as matrix algebra and used for information retrieval.

Collaborative filtering (CF) is a popular approach for recommender systems, and matrix factorization is a well-recognized approach to CF [Koren et al., 2009; Koren, 2010; Paterek, 2007; Takàcs et al., 2007]. Given the ratings to items by users, latent factor models express both items and users on factors inferred from the patterns of ratings. User-item interactions are then represented as inner products of vectors in the new factor space. High correspondence between item and user factors leads to recommendation of an item to a user. Koren [Koren, 2010; 2008] proposes augmentation to standard SVD for effective prediction of ratings in CF.

In CommTrust, our application of SVD is fundamentally different from existing applications. While existing applications of SVD use matrices in the reduced feature space after decomposition to compute similarity among original data objects, CommTrust computes the relative weights of original data objects in the reduced feature space. Noises are removed and correlation of data objects is considered in SVD transformation, and as a result weights computed in the new reduced feature space are more accurate description of weightings of original data objects.

Chapter 3

CommTrust: Comment-based Multi-dimensional Trust Evaluation

We view feedback comments as a source where buyers express their opinions more honestly and openly. Our analysis of feedback comments on eBay and Amazon reveals that even if a buyer gives a positive rating for a transaction, s/he still leaves comments of mixed opinions regarding different aspects of transactions in feedback comments. Table 3.1 lists some sample comments, together with their rating from eBay. For example for comment c_2, a buyer gave a positive feedback rating for a transaction, but left the following comment: "*bad communication, will not buy from again. super slow ship(ping). item as described.*". Obviously the buyer has negative opinion towards the *communication* and *delivery* aspects of the transaction, despite an overall positive feedback rating towards the transaction. We call these salient aspects *dimensions* of e-commerce transactions. Comment-based trust evaluation is therefore multi-dimensional. Hereafter we will use the terms opinion and rating interchangeably to express the positive, negative and neutral polarities toward entities that expressed in natural language text.

3.1 The CommTrust model

Figure 3.1 depicts the CommTrust framework. Unlike existing trust models (including the one used on eBay) where explicit transaction feedback ratings (positive or negative) are used to compute overall trust scores for sellers. Aspect opinion expressions, and their associated ratings (positive or negative) are first extracted from feedback comments. Dimension trust scores together with their weights are further computed by aggregating dimension ratings.

Table 3.1: Sample comments on eBay

No	Comment	eBay rating
c_1	beautiful item! highly recommend using this seller!	1
c_2	bad communication, will not buy from again. super slow ship(ping). item as described.	1
c_3	quick response	1
c_4	looks good, nice product, slow delivery though.	1
c_5	top seller. many thanks. A+	1
c_6	great price and awesome service! thank you!	1
c_7	product arrived swiftly! great seller.	1
c_8	great item. best seller of ebay	1
c_9	slow postage, didn't have the product asked for, but seller was friendly.	1
c_{10}	wrong color was sent, item was damaged, did not even fit phone.	1

Note: 1 = Positive, 0 = Neutral, -1 = Negative

The algorithm for mining feedback comments for dimension ratings and the technique for computing dimension weights will be described in Chapter 4 and Chapter 5.

Definition 3.1.1. The overall trust score T for a seller is the weighted aggregation of dimension trust scores for the seller,

$$T = \sum_{d=1}^{m} t_d * w_d, \tag{3.1}$$

where t_d and w_d represent respectively the trust score and weight for dimension d ($d = 1..m$).

The trust score for a dimension is the degree or probability that buyers express positive opinion towards the dimension, and roughly is positively correlated with the proportion of positive ratings towards the dimension. However, buyers only express limited positive or negative opinions towards some dimensions in feedback comments. Computing the trust score from a limited number of samples has a high chance of over estimate. For example, out of 1,956 feedback comments for ten eBay sellers of the transactions from 31 January to 18 March 2012, only 73 comments contain ratings towards the *communication* dimension, where 72 are positive. The percentage of positive ratings is thus 98.6% (72 out of 73). However, this estimation is made from a limited sample of only 73 ratings. We propose to apply Bayesian adjustment to compute the trust scores for dimensions from a limited number of ratings.

Following the definition of trust by Jøsang et al. [2007], the trust score on a dimension for a seller is the probability that buyers expect the seller to carry out transactions on this

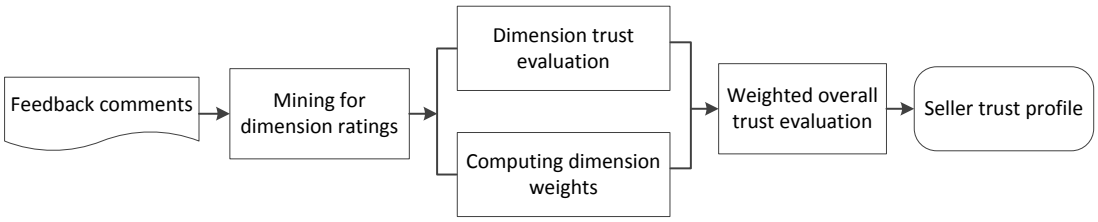


Figure 3.1: The CommTrust framework

dimension satisfactorily. The trust score for a dimension can be estimated from the number of observed positive and negative ratings towards the dimension. Let S=$\{X_1, ..., X_n\}$ be n observations of binary positive and negative ratings, where y observations are positive ratings. S follows binomial distribution $B(n, p)$. Following the Bayes rule, p can be estimated from observations and some prior probability assumption. Assuming the Beta distribution for the prior,

$$Beta(p|\alpha, \beta) == \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}$$

where α and β are hyper-parameters expressing prior beliefs, the Bayes estimate of p is formed by linearly combining the mean $\alpha/(\alpha+\beta)$ from prior distribution and the mean y/n, as below [Casella and Berger, 1990; Heinrich, 2005]

$$\hat{p} = \frac{y + \alpha}{n + \alpha + \beta}. \tag{3.2}$$

Note that the Beta distribution is a special case of the Dirichlet distribution for two dimensions [Heinrich, 2005].

It has been shown in the Beta reputation system [Jøsang and Ismail, 2002] that the assumption of Beta distribution for the prior belief leads to reasonable trust evaluation. The Beta reputation system adopts constant settings of $\alpha = \beta = 1$ for Equation 3.2. We develop the approach further by introducing hyper-parameter settings for α and β to suit for a varying number of observed positive and negative ratings. It is preferable to have only one parameter for trust evaluation [Jøsang and Ismail, 2002]. With the prior belief of neutral tendency for trust, it can be assumed that $\alpha = \beta$. Let $\alpha + \beta = m$, then $\alpha = \beta = 1/2 * m$. The trust score for a dimension is thus defined as follows:

Definition 3.1.2. Given n positive (+1) and negative (-1) ratings towards dimension d,

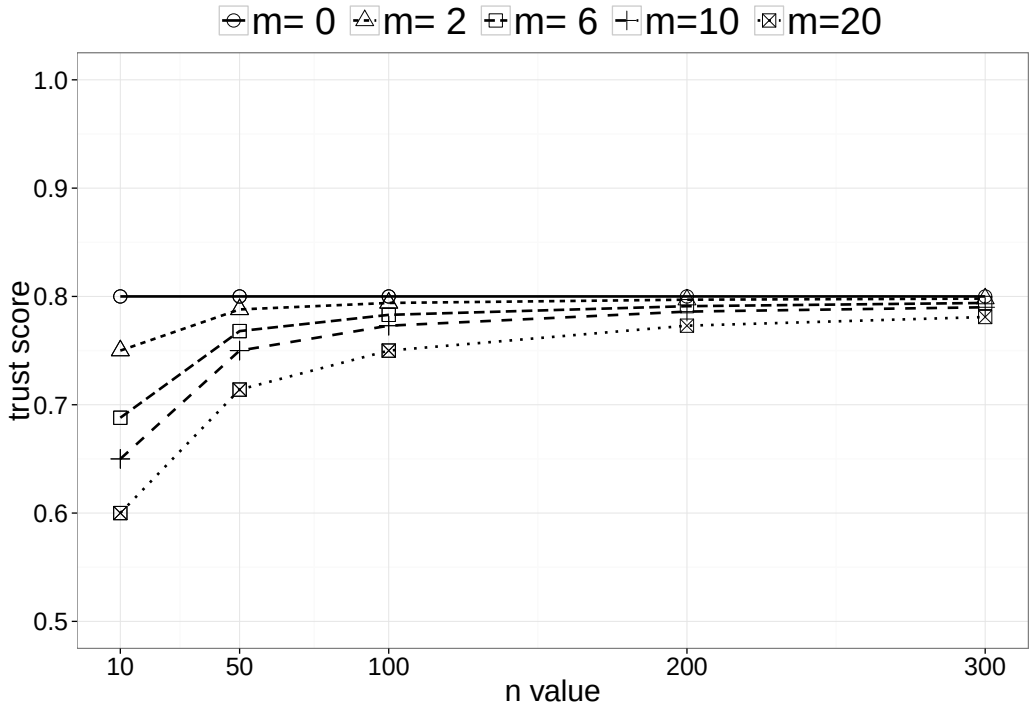


Figure 3.2: The dimension trust score model

$n = |\{v_d | v_d = +1 \vee v_d = -1\}|$, the trust score for d is:

$$t_d = \frac{|\{v_d | v_d = +1\}| + 1/2 * m}{n + m}. \tag{3.3}$$

Equation 3.3 is also called *m-estimate* [Karplus, 1995]. According to Definition 3.1.2, t_d is in the range of [0..1], and 0.5 represents the neutral tendency for trust. In Equation 3.3, m is a hyper-parameter and can be seen as pseudo counts – $1/2 * m$ counts for the positive and negative classes respectively. The higher value of m, the more actual observations are needed to revise the natural neutral trust score of 0.5. More importantly by introducing the prior distribution using the super-parameter m, the adjustment can reduce the positive bias in ratings, especially when there are a limited number of positive and negative ratings [Resnick et al., 2000; Resnick and Zeckhauser, 2002].

Based on our experiment datasets (1,956 feedback comments for ten eBay sellers of the transactions from 31 January to 18 March 2012) refer to Section 5.3, Figure 3.2 plots trust score t_d by Equation 3.3 in relation to different settings of total number of ratings n and pseudo counts m. The figure is plotted for $y/n = 0.8$, and similar trends are observed for

Table 3.2: An example of dimension ratings for a seller on eBay

No	Product	Delivery	Comm.	Cost	Tran.	eBay	Comment
c_1	1	0	0	0	1	1	beautiful item! highly recommend using this seller!
c_2	0	0	0	0	1	1	great service
c_3	0	0	1	0	0	1	quick response
c_4	1	-1	0	0	0	1	looks good, nice product, slow delivery though.
c_5	0	0	0	0	1	1	top seller. many thanks. A+
c_6	0	0	0	1	1	1	great price and awesome service! thank you!
c_7	0	1	0	0	1	1	product arrived swiftly! great seller.
c_8	1	0	0	0	1	1	great item. best seller of ebay
c_9	-1	-1	0	0	1	0	slow postage, didn't have the product asked for, but seller was friendly.
c_{10}	-1	0	0	0	0	-1	wrong color was sent, item was damaged, did not even fit phone.

Comm.: Communication, Tran.: Transaction

other values of y/n. It shows that when the total number of observed ratings n is large ($n \geq 300$), t_d is not very sensitive to the settings of m and converges to the observed positive rating frequency of 0.8. When there is a limited number of observed ratings, that is $n < 300$, an observed high positive rating frequency y/n is very likely an overestimation, and so m is set to regulate the estimated value for t_d. With $m = 2$, when $n \geq 50$ $t_d \approx 0.8$. On the other hand, with $m = 20$, only when $n \approx 300$ $t_d \approx 0.8$. From our experiments, settings of $m = 6..20$ typically give stable results. By default, we set $m = 6$.

Example 3.1.1. Table 3.2 shows dimension ratings from ten feedback comments for a seller randomly selected from eBay. We annotated the commens as five dimensions correspond to the aspects of *Item as described*, *Shipping time*, *Communication* and *Postage and handling charges* of the Detailed Seller Ratings on eBay.[1] In addition, users often express opinions directly towards the overall experience with transactions in comments. For example, a comment like "*Great eBayer. A+++++*" has not commented specific aspects of a transaction but rather describes an overall experience towards the Transaction. The feedback ratings

[1] http://pages.ebay.com.au/help/feedback/detailed-seller-ratings.html.

from eBay are listed in the colomn of "eBay".

Note that out of ten comments, only comment c_3 contains opinion for Communication and only c_6 contains opinion for Cost. This lack of ratings for the Communication and Cost dimensions is indeed widespread on eBay and Amazon.

Following Equation 3.3, setting $m = 6$ for such a small sample, the results of dimension trust scores for the dimensions Product, Delivery, Communication and Cost are 54.55%, 44.44%, 57.14% and 57.14% respectively. It is obvious that there is high variance in the dimension trust scores for this seller, from 44.44% for Delivery to 57.14% for Communication and Cost. The dimension trust scores give a comprehensive trust profile for the seller.

Assuming an equal weight of 0.25 for all dimensions in Equation 3.1, the average of dimension trust scores is 53.3%. Given that the trust score for the Delivery dimension is as low as only 44.44%, the overall trust score of 53.3% is reasonable estimate for the overall trust level for this seller. In contrast, according to the feedback ratings on eBay, the positive feedback percentage is 88.89% (eight positive ratings out of a total of nine non-zero ratings). Obviously the eBay positive feedback percentage score is inflated due to the strong positive bias in limited number of nonzero ratings. On the other hand, the overall trust score from transaction ratings by Equation 3.3 is 76.92% (seven positive ratings out of a total of seven non-zero ratings), which still can not bring down the overall trust score to a reasonable level.

CommTrust can significantly reduce the strong positive bias in eBay reputation systems, and solve the "all good sellers" problem. With CommTrust, seller ratings are regulated to a more reasonable level and truly reputable sellers are effectively differentiated from irreputable sellers.

3.2 A user study

A user study was conducted to elicit users ranking of sellers from reading feedback comments, which was also used as the ground truth for evaluating the CommTrust multi-dimensional trust evaluation model. Inspired by evaluation techniques from the Information Retrieval community [Thomas and Hawking, 2006], experiment participants are asked to judge differences rather than make absolute ratings. For ten sellers, each seller is paired with every other seller and form 45 pairs. The orders for pairs and for sellers within pairs were randomised to avoid any presentational bias. Each pair was judged by five users and a seller preferred by at least three users was seen as a vote for the seller. The total number of preference votes from 45 pairs for each seller were used as the preference score to rank sellers.

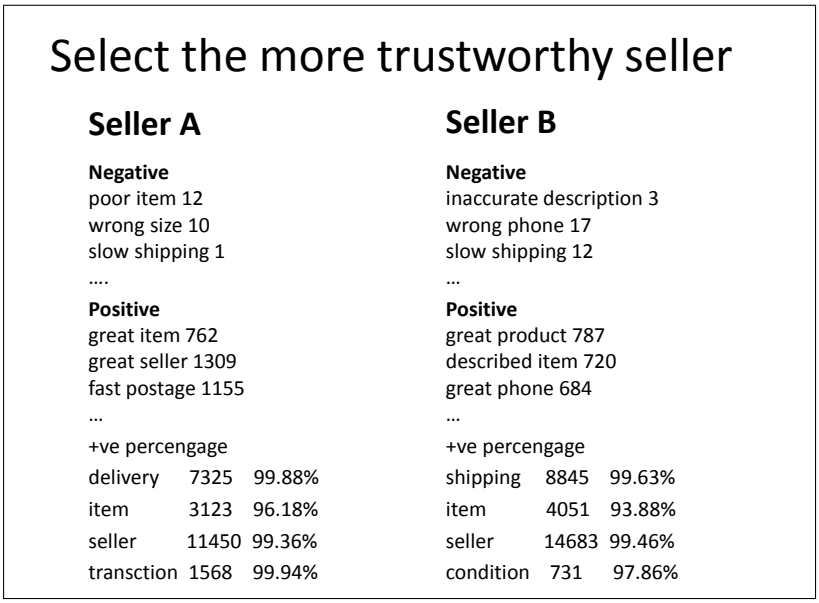


Figure 3.3: A sample of pairwise preference in user study

It is infeasible to ask participants to read all comments for two sellers and choose a preferred seller. We therefore generated summaries of comments for sellers. The comment summaries for each pair of users were presented side by side to elicit users preference judgements. For a seller, we generated opinionated phrases for four dimensions, where positive and negative phrases for each dimension are ordered by decreasing frequency. The three most frequent positive and negative phrases for each dimension formed the summary for a seller. An example page for the survey is shown in Figure 3.3.

In this user study, the degree of agreement among five users annotation is evaluated by the Kappa statistics [Fleiss, 1971]. The following equation shows how the overall value of kappa is calculated:

$$k = \frac{P_o - P_e}{1 - P_e}$$

where P_o denotes the observed agreement, and P_e denotes the expected agreement. The kappa values range from -1 to +1. Landis and Koch [1977] interpreted kappa statistics as shown in Table 3.3. On the eBay data dataset, $k = 0.58$, and on the Amazon dataset, $k = 0.63$. The user annotation in our experiments falls in moderate to substantial agreement.

The ultimate goal of trust evaluation for e-commerce applications is to rank sellers and help users select trustworthy sellers to transact with. In this respect, in addition to absolute trust scores, relative rankings are more important for evaluating the performance of different

Table 3.3: Meanings of the Kappa statistics

Kappa value	Agreement level
< 0	poor agreement
$[0, 0.2)$	slight agreement
$[0.2, 0.4)$	fair agreement
$[0.4, 0.6)$	moderate agreement
$[0.0, 0.8)$	substantial agreement
$[0.8, 1)$	almost perfect agreement

trust models. To this end, we employ Kendall's τ [Sheskin, 2004] to measure the correlation between two rankings based on the number of pairwise swaps that is needed to transform one ranking into another. Kendall's τ measures the degree of two sets of ranks with respect to the relative ordering of all possible pairs. In other words it is the difference between the probability that the observed data are in the same order versus the probability that the ovserved data are not in the same order. The following equation shows how Kendall's τ is the probability of the difference of the concordant pairs and the discordant pairs.

$$\tau = \frac{n_C - n_D}{n(n-1)/2}$$

where:

n_C is the number of concordant pairs of ranks

n_D is the number of discordant pairs of rands

$[n(n-1)]/2$ is the total number of possible pairs of ranks

A concordant pair is when the rank of the second variable is greater than the rank of the former variable. A discordant pair is when the rank is equal to or less than the rank of the first variable.

The value of τ falls in $[-1, 1]$, a positive value indicates positive correlation, zero represents independence and a negative value indicates negative correlation. When $\tau = 1$, indicates the complete agreement among the rankings (i.e., all of the pairs of ranks are concordant), and when $\tau = -1$, indicates the complete disagreement among the rankings (i.e., all of the pairs of ranks are discordant).

τ is the standard metric for comparing information retrieval systems, and it is generally considered that $\tau \geq 0.9$ for a correlation test suggests two system rankings are equivalent. A large value for $|\tau|$ with $p \leq 0.05$ suggests that two rankings are correlated, and a small value for $|\tau|$ with $p > 0.05$ suggests that two rankings are generally independent.

Table 3.4: Seller rankings by reading comments in user studies

eBay seller	eBay rank	Comment rank	Amazon seller	Amazon rank	Comment rank
Seller 1	6	6	Seller 1	5	5
Seller 2	8	5	Seller 2	4	6
Seller 3	10	10	Seller 3	7	2
Seller 4	4	2	Seller 4	1	1
Seller 5	7	3	Seller 5	3	8
Seller 6	5	1	Seller 6	8	7
Seller 7	9	7	Seller 7	2	4
Seller 8	1	9	Seller 8	9	9
Seller 9	2	4	Seller 9	6	3
Seller 10	3	8	Seller 10	10	10
Kendall's τ=0.1111, p-value=0.7275			Kendall's τ=0.4222, p-value= 0.1083		
rank-diff=3			rank-diff=1.8		

Results from the experiment for eBay and Amazon sellers are summarised in Table 3.4. Under the column heading of Comment rank is the ranking of sellers by user preferences after participants read the comment summaries for sellers. The eBay rank and the Amazon rank are based on our experiment data, refer to Section 4.4.1. The eBay rank is calculated based on the total number of positive and negative feedback ratings for transactions in the last 12 months, that is the percentage of positive ratings out of the total number of positive ratings and negative ratings. Note that this score is very close to that computed by Equation 3.3, especially when the sample size is large. The Amazon rank is the ranking of sellers by the average rating in the past 12 months. The correlation between rankings are measured by Kendall's τ. The rank difference between two ranking vectors is defined as:

$$\text{rank-diff} = \frac{\sum_i rank(i) - rank'(i)}{N}$$

where rank(i) and rank'(i) are respectively the rank for seller i by two ranking methods, and N=10. The low Kendall's τ value (0.1111 and 0.4222) and high p-value (0.7275 and 0.1083) suggest that on eBay and Amazon, user preference rankings after reading comment summaries are not strongly correlated with the rankings by the respective eBay and Amazon reputation systems. This suggests that the comments contain distinct information for users to rank sellers. The ranking difference of 3 for ten eBay users between rankings by reading comments and by eBay reputation system suggests that on average there is a difference of 3 ranks for sellers by the two approaches. Similarly for Amazon sellers there is difference of

1.8 ranks on average. Our user study demonstrates that it can be speculated that content of comments can be used to reliably evaluate the trustworthiness of sellers, which is the objective of CommTrust.

3.3 Summary

In this chapter we have proposed comment-based multi-dimensional trust model, computing overall and dimensional turst scores from feedback comments. Based on an example of dimension ratings for an eBay seller, the CommTrust can significantly reduce the strong positive bias in eBay reputation systems and solve the "all good sellers" problem.

We have performed a user study. In this user study, the users are given the summary of comments from 45 pairs of ten different sellers, and they had to choose which seller is more trustworthy. The findings showed that the ranking from the user study is different from the ranking from eBay and Amazon for the same seller.

Given that the user comments are very rich and diverse, we have proposed CommTrust to identify trustworthy and reliable sellers. In the next chapter, more detailed discussion will be presented on how to mine the comments to extract the dimension and dimension ratings from the comments, and the finds of CommTrust Lexical-LDA.

Chapter 4

Lexical-LDA: Mining Feedback Comments for Dimension Rating Profile by Lexical Topic Modelling

Our research questions are focused on how to extract dimension ratings from feedback comments and further aggregating these dimension ratings to compute dimension trust scores. Therefore we proposed CommTrust Lexical-LDA to solve our research questions.

We will first describe our approach based on the typed dependency analysis to extracting dimension expressions and identifying their associated ratings. Topic modelling is a principled approach to group terms of the same topic into one group [Zhai et al., 2011]. We then propose an algorithm based on popular topic modelling method that Latent Dirichlet Allocation (LDA) [Blei et al., 2003] for grouping dimension expressions into dimensions and computing dimension weights. This approach can achieve stable performance across domains, and the features used are more transparent to a human user.

4.1 Extracting aspect expressions and ratings by typed dependency analysis

On e-commerce sites like eBay and Amazon, user feedback comments are generally short, and very often are phrases or short sentences. Our analysis reveal that these short sentences and phrases can be accurately described using *dependency relations* [De Marneffe and Manning, 2008] in natural language parsing. Based on the parsing results of dependency relations, we can identify dimension-rating patterns that describe dimensions and associated ratings.

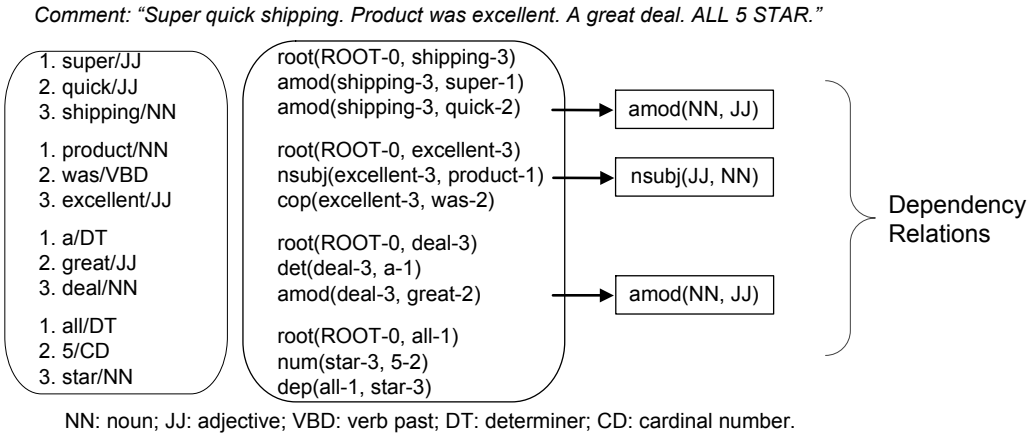


Figure 4.1: Typed dependency relation analysis example

The typed dependency relation representation [De Marneffe and Manning, 2008] is a recent Natural Language Processing (NLP) tool to help understand the grammatical relationships in sentences. All information in a sentence is represented as binary relations between pairs of words while ignoring linguistic details that are not relevant to users. With typed dependency relation parsing, a sentence is represented as a set of dependency relations between pairs of words in the form of (*head, dependent*), where content words are chosen as heads, and other related words depend on the heads. Figure 4.1 shows an example of analysing the comment "*Super quick shipping. Product was excellent. A great deal. ALL 5 STAR.*" using the Stanford typed dependency relation parser. The comment comprises four sentences, and the sentence "*Super quick shipping.*" is represented as three dependency relations. *shipping* does not depend on any other words and is at the root level. The adjective modifier relations *amod (shipping-3, super-1)* and *amod (shipping-3, quick-2)* indicate that *super* modifies *shipping* and *quick* modifies *shipping*. The number following each word (e.g., shipping-3) indicates the position of this word in a sentence. Words are also annotated with their Part of Speech (POS) tags such as noun(NN), verb (VB), adjective (JJ) and adverb (RB).

If a comment expresses opinion towards dimensions then the dimension words and the opinion words should form some grammatical dependency relations expressing the modifying relationship. It has been reported that phrases formed by adjectives and nouns, and verbs and adverbs express subjectivity [Turney, 2002]. Among the dependency relations expressing grammatical relationships, we select the relations that express the modifying relation between adjectives and nouns, and adverbs and verbs, as determined by the dependency relation

Table 4.1: Dimension rating patterns

Dependency Relation	**Patterns**	**Example**
amod(NN, JJ) *adjective modifier*	amod(price/NN, great/JJ) amod(postage/NN, quick/JJ)	*Great price and quick postage, just gorgeous.*
advmod(VB, RB) *adverbial modifier*	advmod(shipping/VB, fast/RB)	*very pretty, fast shipping.*
nsubj(JJ, NN) *nominal subject*	nsubj(prompt/JJ, seller/NN)	*this seller was very prompt.*
acomp(VB, JJ) *adjectival complement*	acomp(arrived/VB, quick/JJ)	*Great CD, arrived quick.*
dep(NN, RB) *dependent*	dep(shipping/NN, fast/RB)	*very fast shipping.*

Note: NN = noun, VB = verb, JJ = adjective, and RB = adverb

parser. Based on our analysis of a sample dataset of 1956 eBay feedback comments for 10 eBay sellers, five types of dependency relations are found to frequently express *dimension rating patterns* (DR-patterns), as listed in Table 4.1. It can be seen that with the modifying relations generally the noun or verb expresses the target concept under consideration whereas the adjective or adverb expresses opinion towards the target concept. With the example comment in Figure 4.1, the dependency relations adjective modifier *amod (NN, JJ)* and normal subject *nsubj (JJ, NN)* suggest DR-patterns *(shipping, super)*, *(shipping, quick)*, *(excellent, product)* and *(deal, great)*. DR-patterns comprise a dimension word and an opinion word. We also call dimension word as *head term*, opinion word as *modifier term*, and the pair of (head term, modifier term) as *dimension expression* or aspect expression. For example, with the DR-pattern *amod (price/NN, great/JJ)*, "price" is the head dimension word while "great" is the dependent opinion word.

Ratings from DR-patterns towards the head terms are identified by identifying the prior polarity of the modifier terms by SentiWordNet [Baccianella et al., 2010], a public opinion lexicon. The prior polarities of terms in SentiWordNet include positive, negative or neutral, which corresponds to the ratings of +1, -1 and 0. More detailed information will be descussed in Section 4.3.

The negation relation (*neg()*) from the Stanford parser is used to detect any negation of ratings in DR-patterns. We go through the dependency relations for DR-patterns for the negation modifier *neg()* of the head or dependent. If either the head or dependent word of a DR-pattern involves the *neg()* relation, the relevant rating is inverted. For example, a buyer of eBay wrote the comment "after five days waiting, the item is not available.". Applying the

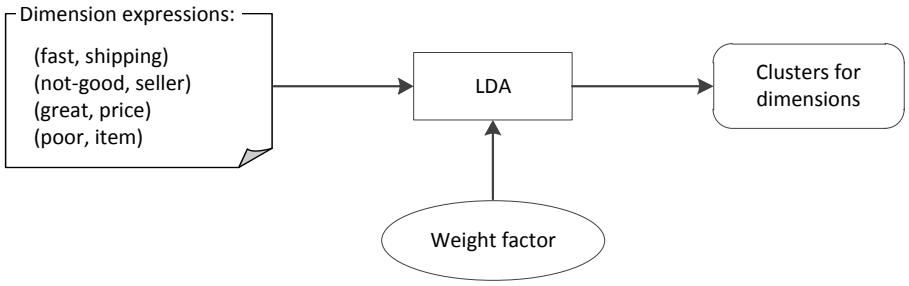


Figure 4.2: The Lexical-LDA framework

typed dependency relation parser, we can get "*num(days-3, five-2)*, *prep_after(available-10, days-3)*, *partmod(days-3, waiting-4)*, *det(item-7, the-6)*, *nsubj(available-10, item-7)*, *cop(available-10, is-8)*, *neg(available-10, not-9)*, *root(ROOT-0, available-10)* ". The POS tags of this comment are " *after/IN, five/CD, days/NNS, waiting/VBG, the/DT, item/NN, is/VBZ, not/RB, available/JJ* ". The suggested pair from DR-patterns is *nsubj(available, item)*. The negation relation *neg(available, not)* modifies the word *available* from suggented pair. The extracted expression is *(item, available, neg)*.

Ratings from dimension expressions towards the head terms are identified by identifying the prior polarity of the modifier terms by SentiWordNet, a public opinion lexicon. The prior polarities of terms in SentiWordNet include positive, negative or neutral, which corresponds to the ratings of +1, -1 and 0. Negations of dimension expressions are identified by the *Neg()* relation of the dependency relation parser. When a negation relation is detected the prior polarity of the modifier term is inverted.

4.2 Grouping dimension expressions into dimensions

For obvious topic modelling is a model for automatic discovering granularity "topics". We propose the Lexical-LDA algorithm to group aspect expressions into semantically coherent categories, which we call *dimensions*. Different from the conventional topic modelling approach, which takes the document by term matrix as input, Lexical-LDA makes use of shallow lexical knowledge of dependency relations for topic modelling to achieve more effective clustering. Figure. 4.2 depicts the Lexical-LDA framework.

We make use of two types of lexical knowledge to "supervise" grouping dimension expressions into dimensions so as to produce meaningful clusters.

- Comments are short and therefore co-occurrence of head terms in comments is not very

informative. We instead use the co-occurrence of dimension expressions with respect to a same modifier across comments, which potentially can provide more meaningful contexts for dimension expressions.

- We observe that it is very rare that the same aspect of e-commerce transactions is commented more than once in the same feedback comment. In other words, it is very unlikely that the dimensions expressions extracted from the same comment are about the same topic.

With the shallow lexical knowledge of dependency relation representation for dimension expressions, the clustering problem is formulated under topic modelling as follows: the dimension expressions for a same modifier term or negation of a modifier term are generated by a distribution of topics, and each topic is generated in turn by a distribution of head terms. This formulation allows us to make use of the structured dependency relation representations from the dependency relation parser for grouping. Input to Lexical-LDA are dependency relations for dimension expressions in the form of *(modifier, head)* pairs or their negations, like *(fast, shipping)* or *(not-good, seller)*.

Gibbs sampling has been proposed as approximate inference for LDA [Griffiths and Steyvers, 2004]. A detailed description of the derivation process for a Gibbs sampler for LDA is given in [Heinrich, 2005], while we only present the results below. Let M, K and V denote respectively the number of documents, the number of topics and the number of word tokens in the vocabulary. Let also that $\vec{\alpha}$ and $\vec{\beta}$ respectively be the hyper-parameters on the mixing proportions for topics and on the mixture components of topics. Equation 4.1 below is the update equation for computing the full conditional distribution of a word token w_i for a topic k, where $i = (m, n)$ denote the n^{th} word in the m^{th} document, $\vec{w} = \{w_i = t, \vec{w}_{\neg i}\}$, $\vec{z} = \{z_i = k, \vec{z}_{\neg i}\}$ and $n^{(.)}_{.,\neg i}$ denote counts, token i is excluded from the corresponding document or topic, and the hyper-parameters are omitted.

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) \propto \frac{n^{(t)}_{k,\neg i} + \beta_t}{\sum_{t=1}^{V}(n^{(t)}_{k,\neg i} + \beta_t)} \cdot \frac{n^{(k)}_{m,\neg i} + \alpha_k}{\sum_{k=1}^{K}(n^{(k)}_{m,\neg i} + \alpha_k)} \tag{4.1}$$

The second type of lexical knowledge that generally two head terms from the same comment are for different dimensions is applied in LDA as a weight factor for adjusting the conditional probability for assigning head terms for a modifier term to dimensions. Specifically, for a head term w_i with index $i = (m, n)$, in computing the conditional probability for assigning w_i to topic k, we consider the evidence as presented by the head terms appearing

in a same comment as w_i: when computing the conditional probability of $p(z_i = k|\vec{z}_{\neg i}, \vec{w})$, head terms in a same document with w_i and is associated with a topic other than k casts a positive vote for the conditional probability as expressed in Equation 4.1 and otherwise a negative vote. The weight factor is thus defined as:

$$f(z_i = k) = \frac{n_{m,\neg i}^{(c,\neg k)} - n_{m,\neg i}^{(c,k)}}{n_{m,\neg i}^{(c)}}$$

where c denotes the set of comments that w_i appears, $n_{m,\neg i}^{(c,\neg k)}$ denotes the count of head terms of m other than w_i that appear in any comment of c and is assigned to a topic other than k, $n_{m,\neg i}^{(c,k)}$ denotes the count of head terms for m other than w_i that appears in any one comment of c and is assigned to topic k, and $n_{m,\neg i}^{(c)}$ denotes the count of head terms for m other than w_i that appear in any comment of c. As a result $f(z_i = k) \in [-1, 1]$, and

$$\begin{cases} > 0 \text{ more positives votes,} \\ = 0 \text{ same number of positive and negative votes,} \\ < 0 \text{ more negative votes.} \end{cases}$$

We apply the weight factor to adjust the computation of conditional probability in Equation 4.1. Given head term w_i with index $i = (m, n)$ – the n^{th} head term for a modifier term m, if there are head terms that appear in the same comment as w_i, Equation 4.1 is adjusted as follows:

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) \propto (1 + \alpha * f(z_i = k)).\frac{n_{k,\neg i}^{(t)} + \beta_t}{\sum_{t=1}^{V}(n_{k,\neg i}^{(t)} + \beta_t)}.\frac{n_{m,\neg i}^{(k)} + \alpha_k}{\sum_{k=1}^{K}(n_{m,\neg i}^{(k)} + \alpha_k)} \tag{4.2}$$

Three cases need to be distinguished when applying Equation 4.2.

- If $f(z_i = k) > 0$, that is there are more head terms in the same comments that support assigning w_i to topic k, the conditional probability estimate by the original Gibbs sampler is increased.
- If $f(z_i = k) < 0$, that is there are more head terms in the same comments that are against the assignment of w_i to topic k, the conditional probability estimate by the original Gibbs sampler is decreased.
- Otherwise $f(z_i = k) = 0$, the original Gibbs sampler estimate is kept.

In Equation 4.2, $\alpha \in [0, 1]$ is a parameter indicating the level of strength of the knowledge encoded in $f(z_i = k)$. The reason is that such knowledge is probabilistic in nature. The adjustment component $(1 + \alpha * f(z_i = k))$ is in the range $[1 - \alpha, 1 + \alpha]$. Note that the adjusted probability computed by Equation 4.2 shall be normalised for all topics afterwards.

The *(modifier, head)* structures are first used for topic discovery in [Lu et al., 2009]. In [Lu et al., 2009] Probabilistic Latent Semantic Analysis (PLSA) is applied where mixing weight for themes (dimensions) are assumed and optimised using the EM procedure. In our formulation the LDA model is used. More importantly we apply further lexical knowledge to constrain the process of clustering head terms to produce more meaningful clusters.

Our application of the second type of lexical knowledge to "supervise" the topic modelling process is motivated by the notion of "cannot links" in [Zhai et al., 2011], although conventional LDA on documents of word tokens is applied there. Their application of constraints at the sentence level potentially can result in a large number of such constraints. In addition to the "cannot-link" constraints, "must-link" constraints are used to state that some phrases with common words likely belong to the same topic. For example "battery power" and "battery life" likely belong to the same topic. Although such phrases may be widespread in product reviews, they are rare in e-commerce feedback comments. It is wroth noting that it is shown in [Zhai et al., 2011] that the cannot-link constraints produce more effectiveness on the clustering results than the must-link constraints.

When *(modifier, head)* pairs and their negations are grouped into dimensions, we compute weights for dimensions. Intuitively the weight for a dimension is proportional to the total number of positive and negative ratings on the dimension. Specifically we compute the total number of *(modifier, head)* dimension expressions for the dimension. Indeed only frequent dimension expressions with head terms appearing in at least 0.1% of comments are included. The total number of dimension expressions for dimensions are normalised to produce the dimension weights.

4.3 Rating evaluation

We apply a general opinion word lexicon SentiWordNet [Baccianella et al., 2010], which is a widely used public domain NLP resource to identify opinion polarities. When *(modifier, head)* pairs are grouped into dimensions, the associated modifier terms express the opinion priority of dimensions.

SentiWordNet has a total of 155,181 words and each word (together with a POS tag) is

annotated with positive, negative and objective scores that are summed to one. The sum of positive and negative scores for a word indicates its level of subjectivity. We apply a threshold of 0.5 for subjectivity – words with a sum of positive and negative scores of at least 0.5 are subjectivity words. Moreover, an opinion word takes on the positive polarity if its positive score is higher, or negative polarity otherwise. In applying SentiWordNet, words with a sum of positive and negative scores greater than or equal to 0.5 is considered as express subjectivity, and if the positive score equals negative score for a subjective word, the word carries prior positive polarity.

It is well known that whether a word expresses opinion and the polarity associated with a word depend on context and vary for domains. SentiWordNet is a general opinion lexicon compiled from several application domains and their word annotations need to be reviewed to be applied to the e-commerce domain. For example, *top* and *prompt* are both labelled as objective and not having any polarities in SentiWordNet, but they indeed express positive opinions in the e-commerce domain, as in expressions like "*top teller*" and "*prompt service*". Based on our analysis of a sample of 1,956 comments for ten eBay sellers, some word polarity labels in SentiWordNet are reviewed as follows:

- Ten objective words are changed to opinion words and annotated with prior polarity, including *lightning (+ve), fast (+ve), prompt(+ve), saftely (+ve), pretty (+ve), satisfied (+ve), scratch (-ve), squashed (-ve), late (-ve)*, and *waste (-ve)*.
- *cheap (-ve)* is re-annotated as *cheap (+ve)*.
- *weak (+ve)* is re-annotated as *weak (-ve)*.

4.4 Experiments

Extensive experiments on two e-commerce datasets and one hotel review datasets were conducted to evaluate various aspects of CommTrust Lexical-LDA, including the dimension grouping algorithm and the turst model. The hotel review dataset is specifically used to demonstrate the generality of Lexical-LDA in domains other than e-commerce.

4.4.1 Datasets

180,788 feedback comments were crawled for ten eBay sellers on ebay.com, where two sellers were randomly selected for each of five categories on the "Shop by category" list on

Table 4.2: The eBay dataset

Seller	Category	#comments	Feedback score	Pos feedback (%)
Seller 1	baby	5876	5481	99.6%
Seller 2	baby	4542	3618	100%
Seller 3	camera	2717	2609	99.4%
Seller 4	camera	27887	26487	99.4%
Seller 5	computer	5596	5457	99.9%
Seller 6	computer	27969	24199	99.9%
Seller 7	jewelry	3628	3194	100%
Seller 8	jewelry	60000	53624	99.7%
Seller 9	phone	34582	33237	99.4%
Seller 10	phone	29082	27392	99.5%

Table 4.3: The eBay dataset of Detailed Seller Ratings

Seller	Pos feedback (%)	Detailed Seller Ratings (#ratings)			
		Item	Comm	Shipping	Cost
Seller 1	99.6%	4.8 (2691)	4.9 (2679)	4.9 (2687)	4.8 (2660)
Seller 2	100%	5 (221)	4.9 (223)	4.8 (223)	4.9 (229)
Seller 3	99.4%	4.9 (832)	4.9 (829)	4.9 (837)	5 (919)
Seller 4	99.4%	4.9 (12034)	4.9 (13046)	4.9 (12653)	5 (14019)
Seller 5	99.9%	5 (4803)	4.9 (4998)	4.9 (4795)	5 (5299)
Seller 6	99.9%	4.9 (15505)	4.9 (15934)	4.9 (15438)	5 (17679)
Seller 7	100%	4.9 (925)	5 (986)	5 (961)	4.9 (920)
Seller 8	99.7%	4.9 (44095)	5 (47734)	4.9 (45622)	5 (48088)
Seller 9	99.4%	4.9 (3983)	5 (4375)	4.9 (4402)	5 (4717)
Seller 10	99.5%	4.9 (5940)	5 (6507)	4.9 (6453)	5 (6929)

eBay.com, including *Cameras & Photography*, *Computers & Tablets*, *Mobile Phones & Accessories*, *Baby*, and *Jewellery & Watches*. Note that the sellers also sell products in other categories in addition to the listed categories. For evaluation of our trust model, the feedback profile for each seller were also extracted [1]:

- *The feedback score* is the total number of positive ratings for a seller from past transactions.
- *The positive feedback percentage* is calculated based on the total number of positive and negative feedback ratings for transactions in the last 12 months, that is $\frac{\#\text{positive-ratings}}{\#\text{positive-ratings}+\#\text{negative-ratings}}$.

[1] `pages.ebay.com/services/forum/feedback.html`.

Table 4.4: The Amazon dataset

Seller	Category	#comments	Avg. rating
Seller 1	Electronics-Computer	4365	4.8
Seller 2	Electronics-Computer	4786	4.8
Seller 3	Electronics-Camera	3202	4.9
Seller 4	Electronics-Camera	8000	4.8
Seller 5	Electronics-Phone	5097	4.7
Seller 6	Electronics-Phone	2631	4.8
Seller 7	Jewelry-Ring	6281	4.6
Seller 8	Jewelry-Ring	1295	4.5
Seller 9	Baby-Tub	3860	4.8
Seller 10	Baby-Diaper	927	4.7

- *The Detailed seller ratings* of a seller are five-star ratings on the following four aspects: *Item as described (Item)*, *Communication (Comm)*, *Shipping time (Shipping)* and *Shipping and handling charges (Cost)*. The DSR profile shows a sellers average rating and the number of ratingss. Average ratings are computed on a rolling 12-month basis, and will only appear when at least ten ratings have been received.

Details of the dataset are as shown in Table 4.2, the detailed seller ratings of eBay sellers are shown in Table 4.3.

On Amazon, for a third-party seller, an average rating in the past 12 months is displayed, together with the total number of ratings. Each rating is associated with a short comment. 40,444 comments for ten third-party sellers with a large number of ratings were crawled from five categories, including *Electronics-Computer*, *Electronics-Camera*, *Electronics-Phone* *Jewelry-Ring*, and *Baby-Tub* and *Baby-Diaper*. Note that these sellers also sell products in other categories. A summary of the Amazon dataset is as shown in Table 4.4.

As shown in Tables 4.3 and 4.4, the strong positive bias is clearly demonstrated on the eBay and Amazon datasets. On the eBay dataset, the positive feedback percentage as well as DSR five-star rating scores have little dispersion and can hardly be used by itself to rank sellers. Similarly on the Amazon dataset, the average ratings for six sellers are 4.8 or 4.9.

The TripAdvisor dataset is taken from `http://sifaka.cs.uiuc.edu/~wang296/Data/index.html`, which was originally used in [Wang et al., 2011] and [Wang et al., 2010]. The dataset contains hotel reviews, as well as overall ratings and ratings on seven pre-defined aspects in each review. This dataset was mainly used to evaluate the applicability of Lexical-LDA for dimension grouping in domains other than e-commerce. 246,399 reviews were in

the original dataset and the following preprocessing was applied: reviews with any missing aspect rating or with less than 50 words were removed so that all reviews have coverage of all aspects. Reviews that Stanford parser can not parse were also removed. After pre-processing we have a total of 52,805 reviews.

4.4.2 Evaluation metrics

We employ metrics Rand index (RI) [Rand, 1971; Zhai et al., 2011] and *Clustering Accuracy* (Acc)[Lu et al., 2009] to evaluate the performance of dimension grouping algorithms.

Rand Index [Rand, 1971] is the evaluation measure for clusterings comparison, which is to calculate the fraction of correctly classified pairs to all possible pairs. Rand Index measures both within-cluster and between-cluster agreement of two clustering algorithms. Given an n elements set $V = \{e_1, e_2, \ldots, e_n\}$, suppose $H = \{h_1, h_2, \ldots, h_r\}$ and $L = \{l_1, l_2, \ldots, l_s\}$ represent two partitions generated by different clustering algorithm. Each partition is a subset of V.

Given a pair of elements $x \in V$ and $y \in V$, let $h(x, y)$ and $l(x, y)$ denote respectively the decision by H and L on whether x and y should be grouped into the same cluster.

There are four types of clusterings for the pair (x, y):

- T_1: (x, y) are grouped into the same cluster in H and into the same cluster in L;
- T_2: (x, y) are grouped into the different cluster in H and into the different cluster in L;
- T_3: (x, y) are grouped into the same cluster in H and into the different cluster in L;
- T_4: (x, y) are grouped into the different cluster in H and into the same cluster in L.

T_1 and T_2 are typically interpreted as agreements in the classification of the elements from a pair, T_3 and T_4 represent disagreements. Let $\theta(h(x, y), l(x, y))$ represents the agreement of classification, for any pair of elements from V, the total number of agreements is $\sum_{x \in V} \sum_{y \in V} \theta(h(x, y), l(x, y))$ among the $\binom{n}{2}$ distinct pairs. We can show that

$$RI(H, L) = \frac{\sum_{x \in V} \sum_{y \in V} \theta(h(x, y), l(x, y))}{\binom{n}{2}}$$

$$= \frac{\sum_{x \in V} \sum_{y \in V} \theta(h(x, y), l(x, y))}{|V| \times (|V| - 1)/2}$$

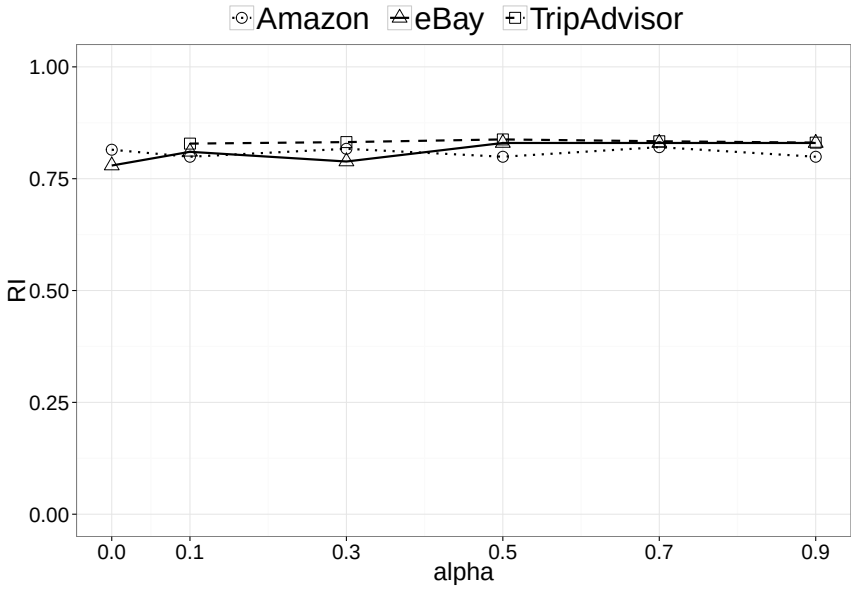


Figure 4.3: The RI of Lexical-LDA dimension grouping

The value of RI falls in [0, 1]. When RI = 0, it indicates the complete disagreement between the clustering algorithms on any pair of elements, and when RI = 1 indicates the complete agreement between the clustering methods.

Acc measures the level of consistency between clusters produced by a clustering algorithm and the clusters by human annotation. Given a set of head terms V, consider a clustering by algorithm H and clustering by human annotation L. Each cluster $C_i(i = 1..k)$ of H is mapped to the cluster of L with the largest number of matching head terms. Let N_i denote the number of head terms in C_i with a matching head term in its corresponding cluster in L. The Acc of H is defined as

$$Acc(H) = \frac{\sum_i^k N_i}{|V|}$$

We employ Kendall's τ [Sheskin, 2004] to evaluate the correlation between the CommTrust rankings and the previous user study rankings that we conducted in Chapter 3.

4.4.3 Evaluation of Lexical-LDA

Informal language expressions are widely used in feedback comments. Some pre-processing was first performed. Spelling correction was applied. Commonly used informal expressions including *A+++* and *thankx* were replaced with *AAA* and *thanks*. The Stanford depen-

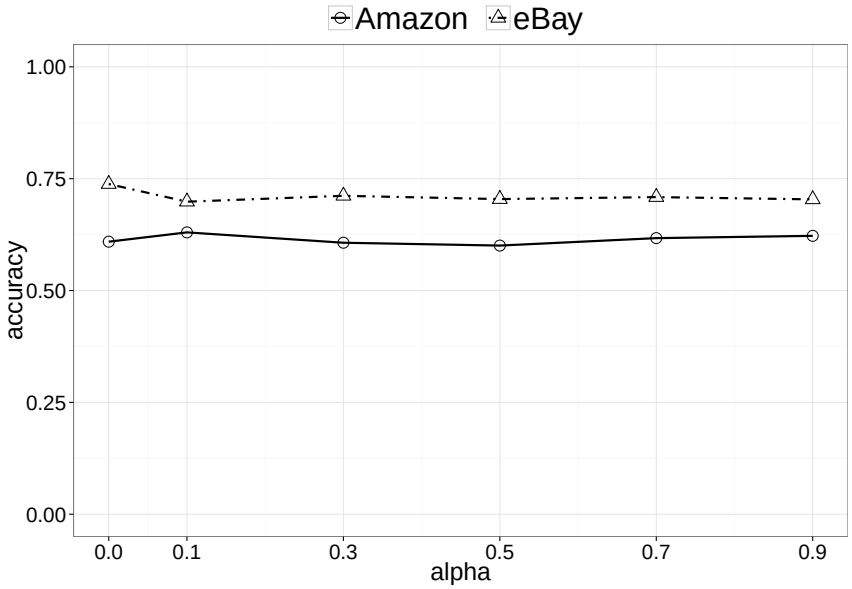


Figure 4.4: The accuracy of Lexical-LDA dimension grouping

dency relation parser was then applied to produce the dependency relation representation of comments and dimension expressions were extracted. The dimension expressions were then grouped to dimensions by the Lexical-LDA algorithm.

To evaluate Lexical-LDA, the ground truth for clustering was first established. Dimension expressions are *(modifier, head)* pairs, and to remove noise only those pairs with support for head terms of at least 0.1% or three comments (whichever is larger) were considered for manual grouping. Some head terms resulted from parsing errors that do not appear to be an aspect were discarded. Examples of such terms include *thanks*, *ok* and *A+++*, In the end a total of maximum 100 head terms were manually grouped based on the inductive approach to analysing qualitative data [Thomas, 2006]. We first grouped head terms into categories according to their conceptual meaning – some head terms may belong to more than one category, and some orphan words were discarded. We then combined some categories with overlapping head terms into a broader category, until some level of agreement was reached between annotators. [2] As a result of this manual labelling process for the eBay and Amazon dataset, the feedback comments for each seller finally seven clusters are obtained.

Lexical-LDA was implemented based on the Mallet topic modelling toolkit [McCallum, 2002]. With dimension expressions in the form of *(modifier, head)* pairs, the modifier term

[2]Manual grouping was performed by two persons. Inconsistency was resolved by discussion.

Table 4.5: The head term clusters for dimensions

Dim	**Manual grouping**	**Lexical-LDA (α=0.5)**	**Standard LDA**
1	item, bag, product, dress, earrings, outfit, top, ring, shoes, coat, necklace, jacket, stuff, one, curtains, handbag, boots, zip, toy, backpack, suit, material, goods, piece, scarf, leggings	item: 532, bag: 146, dress: 70, earrings: 49, outfit:45, coat: 16, top: 16, ring:14, one:11, shoes: 11, jacket:11, necklace: 11, tfit: 8, handbag: 7, look: 7, received: 7, goods: 6, scarf: 3, product: 3	item: 341, bag: 199, dress: 74, earrings: 61, outfit: 50, shoes: 17, coat: 16, ring: 15, necklace: 13, jacket: 11, one: 10, look: 10, curtains: 8, fit: 7, handbag: 6, suit: 6, received: 6, track: 5, toy: 3, piece: 3, leggings: 3, scarf: 3
2	quality, condition, look, size, color, description, fit, described, design	look: 16, size: 10, material: 10, curtains: 8, color: 8, zip: 6, design: 4	size: 11, refund: 8, material: 8, zip: 5, color: 5, design: 5, order: 4, business: 4, post: 3
3	delivery, shipping, postage, dispatch, time, arrived, received, post, shipment, arrival, came	delivery: 1139, payment: 179, shipping: 69, response: 59, postage: 50, dispatch: 25, despatch: 18, deal: 10, came: 10, arrival: 7, arrived: 6, shipment: 5, post: 5	delivery: 1096, shipping: 60, response: 58, postage: 45, dispatch: 22, despatch: 18, deal: 10, came: 10, arrival: 7, arrived: 6, shipment: 5
4	seller, ebayer	seller: 286, ebayer: 286, bayer: 5, described: 4, leggings: 3, track: 3	seller: 519, ebayer: 409, service: 249, communication: 149, product: 138, price: 44, quality: 39, value: 39, buy: 29, condition: 19, looks: 16, top: 15, items: 13, purchase: 13, ebay: 12, time: 11, bayer: 8, stuff: 7, described: 5, boots: 4, description: 4, backpack: 4
5	service, response, track, communication	communication: 142, service: 133, product: 106, quality: 55, price: 46, value: 40, buy: 29, condition: 28, ebay: 11, time: 10, stuff: 8, purchase: 6, boots: 5, description: 5, backpack: 5	goods: 5
6	transaction, buy, deal, purchase, order, business	transaction: 165	transaction: 160
7	payment, price, value, refund	refund: 12, order: 6, business: 5, suit: 4, toy: 4, piece: 1	payment: 147

by head term matrix formed the input for Lexical-LDA. In constructing the cannot-link head term list for a head term (c.f. Section 4.2), only head terms appearing together with the head term in at least 0.1% of or three (whichever is larger) comments were considered. The purpose was to remove the otherwise many spurious cannot-link head terms. The Lexical-LDA parameter settings were: prior pseudo counts for topics and terms were set as $\alpha_k = 0.1$ and $\beta_t = 0.01$ (See Equation (4.2)), the number of topics $K = 4$, 7, 10 for evaluating the trust model and number of iterations was set to 1000.

We evaluate Lexical-LDA against standard LDA for grouping and against the human grouping result. As there are seven categories by human grouping, $K = 7$ for Lexical-LDA. Figure 4.3 plots the RI of Lexical-LDA at different settings of α. Note that the data point for $\alpha = 0$ corresponds to the standard LDA. In addition to the eBay and Amazon datasets, to demonstrate the generality of our approach, the performance of Lexial-LDA on the TripAdvisor dataset is also plotted. For eBay and Amazon data, each plotted data point is the average for ten sellers. On eBay data, RI of Lexical-LDA hovers over $0.78 \sim 0.83$, and Lexical-LDA significantly outperforms standard LDA for $\alpha > 0$ except $\alpha = 0.3$ (p-value < 0.05, paired two-tail t-test). Comparable RI is observed on TripAdvisor and Amazon datasets. Our experiment results indicate that Lexical-LDA has steady performance across different domains.

Figure 4.4 plots the accuracy of Lexical-LDA with different settings of α. As can be seen in the graph, accuracies hover over $0.70 \sim 0.74$ on eBay data and $0.61 \sim 0.63$ on Amazon data. There are not statistically significant differences in accuracies between Lexical-LDA with $\alpha > 0$ and standard LDA, on either Amazon or eBay datasets. However clustering accuracy only measures how automatic grouping matches the human grouping, rather than the coherence within clusters by grouping algorithms. Table 4.5 shows the clusters of head terms for seven dimensions for eBay Seller 1 from manual clustering, Lexical-LDA ($\alpha = 0.5$) and standard LDA respectively. Each head term is grouped to the dimension with the highest frequency. We can see that Lexical-LDA has significantly higher within-cluster coherence than standard LDA. For example Dimension 2 is about the details of items, including for example *quality*, *condition*, *look*, *size* and *colour*. All head terms from Lexical-LDA in this dimension (arguably excluding *curtains*) are indeed about items sold by the seller, although some details are missing. In comparison, the head terms in this dimension from standard LDA are very dispersed and some are not related to items at all, including *refund*, *order*, *business*, and *post*. We believe that the supervision from non-link constraints for head terms helps to produce the meaningful clusters for head terms.

Table 4.6: The precision of identifying different ratings

	Positive	**Negative**	**Neutral**	**Average**
eBay	0.86±0.03	0.60±0.03	0.94±0.02	0.80±0.18
Amazon	0.94±0.03	0.68±0.11	0.93±0.02	0.85±0.15

Table 4.7: Overall trust scores by CommTrust Lexical-LDA for ten eBay sellers

eBay seller	Comment rank	4 dims		7 dims		10 dims	
		trust	rank	trust	rank	trust	rank
Seller 1	6	0.9798	7	0.9777	8	0.9766	7
Seller 2	5	0.9865	1	0.9848	2	0.9828	2
Seller 3	10	0.9771	9	0.9741	9	0.9700	10
Seller 4	2	0.9837	4	0.9836	3	0.9824	3
Seller 5	3	0.9852	2	0.9824	4	0.9824	4
Seller 6	1	0.9850	3	0.9855	1	0.9851	1
Seller 7	7	0.9798	8	0.9783	7	0.9743	8
Seller 8	9	0.9717	10	0.9732	10	0.9725	9
Seller 9	4	0.9823	5	0.9818	5	0.9805	6
Seller 10	8	0.9807	6	0.9814	6	0.9819	5
Kendall's τ			0.6000		0.6889		0.7333
p-value			0.0167		0.0047		0.0022

SentiWordNet is used to decide the prior orientation of modifier terms. Table 4.6 lists the precision of our approach for idenfying positive, negative and neutral ratings on the eBay and Amazon datasets respectively. Precision is calculated as the proportion of correctly identified from all *(modifier, head)* pairs computed for each polarity of positive, negative and neutral. It can be seen that generally our approach achieves reasonably good average precision for all types of ratings — 0.80 ± 0.18 on eBay data and 0.85 ± 0.15 on Amazon data respectively. However the precision for the negative ratings is low, which is mainly due to that SentiWordNet is a general lexicon and as a result some word polarity annotation does not suit the e-commerce application. For example *short* is annotated as neutral and negative in SentiWordNet, and using the latter annotation leads to wrong decision for our application. The problem of adapting general opinion lexicons to different domains is an interesting problem outside the scope of this paper, and readers are referred to the relevant literature (e.g., [Fahrni and Klenner, 2008; Blitzer et al., 2007]).

Table 4.8: Overall trust scores by CommTrust Lexical-LDA for ten Amazon sellers

Amazon seller	Comment rank	4 dims		7 dims		10 dims	
		trust	rank	trust	rank	trust	rank
Seller 1	5	0.8876	6	0.8887	4	0.8861	4
Seller 2	6	0.8924	3	0.8957	3	0.8945	3
Seller 3	2	0.9259	2	0.9223	2	0.9201	2
Seller 4	1	0.8896	5	0.8875	5	0.8837	5
Seller 5	8	0.8750	8	0.8718	8	0.8597	8
Seller 6	7	0.8899	4	0.8857	6	0.8834	6
Seller 7	4	0.8787	7	0.8832	7	0.8791	7
Seller 8	9	0.8643	9	0.8573	9	0.8516	9
Seller 9	3	0.9360	1	0.9317	1	0.9302	1
Seller 10	10	0.7855	10	0.7871	10	0.7961	10
Kendall's τ			0.5556		0.6000		0.6000
p-value			0.0286		0.0167		0.0167

4.4.4 Evaluation of the trust model

Table 4.7 and Table 4.8 list the CommTrust overall trust scores for ten eBay sellers and ten Amazon sellers for 4,7 and 10 dimensions respectively. As the ground truth, the rankings by reading comment summaries for sellers are also listed (under the heading Comment rank). For both eBay and Amazon sellers, on all 4, 7 and 10 dimensions, the rankings by CommTrust are strongly correlated with the ground truth rankings, as demonstrated by the high Kendall's τ and low p-values (less than 0.05). This is suggesting that CommTrust has computed the dimension ratings from comments and they match users' preferences after reading the comments. The number of dimensions does not affect how well the trust scores are correlated with the user rankings.

A strength of CommTrust is that the relative weights that users have placed on different dimensions in their feedback comments can be inferred. However, it is hard to elicit the weights from users when they write the feedback comments. We therefore evaluate our dimension weight prediction indirectly. To verify the effectiveness of the dimension weights in the overall trust score, we compute the unweighted overall trust scores for sellers, and compare the ranking of sellers by unweighted overall trust scores with the ground truth ranking by users. The results of unweighted overall trust scores for eBay and Amazon sellers, on all 4, 7 and 10 dimensions, are shown in Table 4.9 and Table 4.10. It can be seen that without weightings for dimensions, the trust scores for sellers are not correlated with the ground truth ranking of sellers, as demonstrated by low Kendall's τ with all p-value greater

than 0.05.

Table 4.9: Unweighted overall trust scores for ten eBay sellers

eBay seller	Comm rank	4 dims		7 dims		10 dims	
		trust	rank	trust	rank	trust	rank
Seller 1	6	0.9785	5	0.9433	6	0.8712	10
Seller 2	5	0.9811	3	0.9039	9	0.9280	6
Seller 3	10	0.9224	10	0.9563	3	0.8880	9
Seller 4	2	0.9794	4	0.9728	1	0.9655	2
Seller 5	3	0.9243	9	0.9253	7	0.9178	8
Seller 6	1	0.9820	1	0.9042	8	0.9521	3
Seller 7	7	0.9819	2	0.9492	5	0.9296	5
Seller 8	9	0.9690	7	0.9535	4	0.9416	4
Seller 9	4	0.9571	8	0.9618	2	0.9204	7
Seller 10	8	0.9691	6	0.8976	10	0.9689	1
Kendall's τ			0.3333		-0.0667		0.0667
p-value			0.2164		0.8618		0.8618

Table 4.10: Unweighted overall trust scores for ten Amazon sellers

Amazon seller	Comm rank	4 dims		7 dims		10 dims	
		trust	rank	trust	rank	trust	rank
Seller 1	5	0.8543	8	0.8488	2	0.8194	4
Seller 2	6	0.8959	2	0.8673	1	0.8557	1
Seller 3	2	0.8883	3	0.8252	5	0.8021	8
Seller 4	1	0.8563	7	0.8229	7	0.8228	3
Seller 5	8	0.8410	9	0.8283	4	0.7665	9
Seller 6	7	0.8691	5	0.7866	9	0.8063	7
Seller 7	4	0.8971	1	0.8233	6	0.8233	2
Seller 8	9	0.8579	6	0.8168	8	0.8125	6
Seller 9	3	0.8865	4	0.8361	3	0.8153	5
Seller 10	10	0.7387	10	0.6949	10	0.6456	10
Kendall's τ			0.3778		0.2000		0.3333
p-value			0.1557		0.4843		0.2164

The dimension trust scores and weights together form the dimensional trust profiles for sellers. The dimensional trust profiles for ten eBay sellers for four dimensions are shown in Table 4.11. Note that the four dimensions discovered by CommTrust for a seller are the statistically important dimensions that users expressed opinions on in their feedback comments and may not necessarily correspond to the four aspects as specified by eBay DSR ratings. Nevertheless *item* and *shipping* indeed are the dimensions where users comment the

Table 4.11: The dimensional trust profiles for ten eBay sellers

Seller	Dim 1	Weight	Dim 2	Weight	Dim 3	Weight	Dim 4	Weight	Overall
seller 1	0.995	0.306	0.959	0.036	0.961	0.282	0.984	0.377	0.980
seller 2	0.983	0.235	0.976	0.245	0.969	0.015	0.994	0.505	0.987
seller 3	0.986	0.347	0.907	0.023	0.951	0.129	0.981	0.502	0.977
seller 4	0.948	0.113	0.991	0.054	0.985	0.488	0.992	0.345	0.984
seller 5	0.989	0.549	0.935	0.073	0.986	0.036	0.990	0.342	0.985
seller 6	0.947	0.148	0.988	0.483	0.997	0.326	0.996	0.043	0.985
seller 7	0.980	0.183	0.971	0.401	0.988	0.141	0.988	0.275	0.980
seller 8	0.994	0.208	0.978	0.441	0.958	0.325	0.863	0.027	0.972
seller 9	0.994	0.347	0.956	0.006	0.989	0.509	0.927	0.139	0.982
seller 10	0.973	0.018	0.928	0.159	0.989	0.505	0.995	0.319	0.981

Note: The dimensional trust scores and weights for the *item* dimension are underlined.

Table 4.12: The dimensional trust profiles for ten Amazon sellers

Seller	Dim 1	Weight	Dim 2	Weight	Dim 3	Weight	Dim 4	Weight	Overall
seller 1	0.6898	0.1556	0.9462	0.4754	0.9704	0.1955	0.8109	0.1736	0.8876
seller 2	0.9624	0.1884	0.9427	0.3540	0.9249	0.1897	0.7535	0.2678	0.8924
seller 3	0.8571	0.1560	0.9663	0.2008	0.9702	0.5202	0.7597	0.1230	0.9259
seller 4	0.7981	0.1388	0.7241	0.1645	0.9405	0.4960	0.9627	0.2007	0.8896
seller 5	0.9377	0.5294	0.8208	0.1509	0.9288	0.1524	0.6766	0.1674	0.8750
seller 6	0.8286	0.1299	0.9662	0.1675	0.9571	0.4789	0.7243	0.2237	0.8899
seller 7	0.9738	0.1307	0.8744	0.1030	0.9415	0.3450	0.7987	0.4213	0.8787
seller 8	0.9222	0.3535	0.9167	0.1086	0.7573	0.1371	0.8355	0.4007	0.8643
seller 9	0.9841	0.2231	0.8473	0.1555	0.7500	0.0688	0.9646	0.5526	0.9360
seller 10	0.6545	0.1439	0.9136	0.2336	0.9268	0.4103	0.4600	0.2123	0.7855

Note: The dimensional trust scores and weights for the *item* dimension are underlined.

most on. In Table 4.11 the dimensional trust score and weight for the *item* dimension has been underlined. It can be seen that users have substantially different ratings on the *item* dimension for different sellers and put on different weights.

Table 4.12 lists the dimensional trust profiles for ten Amazon sellers. The dimensions *item*, *shipping* and *seller (service)* are the three "hot" dimensions for feedback comments across ten sellers. The fourth dimension includes topics like *condition*, *price* or *packaging*. Generally compared with the eBay dataset, dimensional trust scores are more dispersely distributed among the ten Amazon sellers. The first two columns of Table 4.12 list the dimensional trust scores and weights for the *item* dimension. Obviously the ten sellers are significantly different – trust scores vary from 0.6545 for Seller 10 to 0.9738 for Seller 7, whereas weights vary from 0.1299 for Seller 6 to 0.5294 for Seller 5.

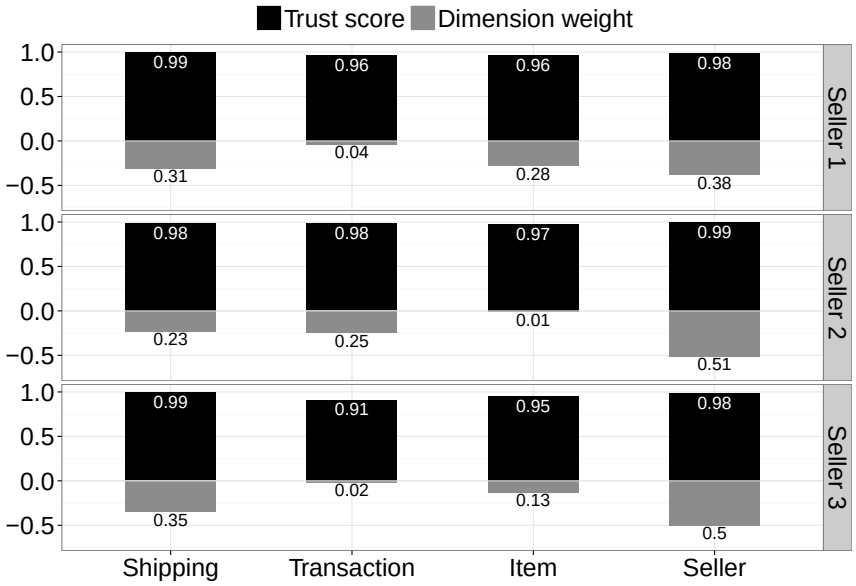


Figure 4.5: The dimension trust profiles by CommTrust Lexical-LDA for sellers

Figure 4.5 depicts the dimensional trust profiles for three eBay sellers Sellers 1, Seller 2 and Seller 3, where they have the same four dimensions, including *shipping*, *cost/response*, *item* and *seller*. For each seller, the upward bars represent trust scores for dimensions while the downward bars represent their weights. For example while having a high overall trust score of 0.9771, Seller 3 has a low dimension trust score of 0.9067 for the *response* dimension (Dimension 2). The figure clearly illustrates the variation of dimension trust for each seller horizontally and those across different sellers vertically. Such comprehensive trust profiles certainly can cater to users preferences for different dimensions and guide users in making informed decisions when choosing sellers.

4.5 Summary

In this chapter, we have answered research questions on how to identify dimension from feedback comments and how to evaluate the weights of each dimension by topic modelling approach.

We have proposed the Lexical-LDA approach for mining feedback comments and trust computation. Lexical-LDA makes use of two types of lexical knowledge based on dependency relations for grouping dimension expressions into dimensions so as to produce meaningful cluster. The weight for a dimension is computed as the total number of dimension expressions

for each dimension.

The findings from the experiments on evaluating the overall trust score similar to the results from the previous user study that we conducted, as explained in Chapter 3. This shows that our proposed trust model is efficient to rank sellers and identify a trustworthy seller. Based on the experiments, the more dimensions used in Lexical-LDA, the better results become. Moreover, the accuracy and Rand Index are conducted to evaluate the grouping in Lexical-LDA on three datasets, eBay, Amazon and TripAdvisor. This shows our algorithm can achieve reasonable results on other than e-Commenton domain.

We performed another experiment without dimensional weighting and the findings are not as good as the CommTrust model. This shows that the dimensional weight is efficient in computing the trust model.

Chapter 5

DR-mining: Lexical Knowledge-based Dimension Rating Profile Analysis

Give dimensions defined by the seed dimension words, we develop a Dimension Rating mining algorithm that incorporates domain knowledge, meta-data, and general grammatical patterns to accurately identifying dimension rating expressions from feedback comments.

We will first describe our approach based on the typed dependency anlaysis to extracting dimension opinion expressions and identifying their associated ratings. We then propose a matrix factorisation technique to automatically compute weights for dimensions from the sparse and noisy dimension rating matrix.

5.1 Knowledge based dimension opinion exraction

We describe our approach of combining meta-data and knowledge base formed by NLP [De Marneffe et al., 2006; De Marneffe and Manning, 2008] techniques to mining textual comments for identifying dimensions and ratings towards dimensions. To more accurately identify dimensions, we further extract meta-data on the product hierarchy from eBay to identify ratings on the product dimension.

Based on eBay Detailed Seller Ratings on four aspects, we define five dimensions:

- *Product*: the quality or condition (new or used) of the product bought.
- *Delivery*: delivery is on time or not.

- *Communication*: how the seller communicates with buyers.
- *Cost*: item price, handling charges, and other associated cost.
- *Transaction*: the overall satisfaction of the transaction.

We need to infer dimension words from their associated prior dimensions. We observe that in eBay and Amazon feedback comments and generally for e-commerce systems salient words and phrases are used to express dimensions and opinions on dimensions. On our sample dataset, out of 751 comments on the Product dimension, 189 comments (25.17%) use the word *product(s)* and 163 comments (21.70%) use the word *item(s)*. We therefore compile the dimension word lists by analysing 1,956 feedback comments from 31 January to 18 March 2012 on ten sellers randomly selected from eBay. Each comment is annotated with the dimensions and their associated opinion as positive (+1), negative (-1) or nil (0). [1] In example of one eBay comment "*bad communication, will not buy from again. super slow ship(ing). item as described.*", positive rating is expressed towards the product dimension, negative rating is expressed towards the communication and delivery dimenions, and no rating is expressed towards any other dimensions.

Table 5.1: Dimension words

Dimension	Words (Confidence)
Product	product (0.99), condition (0.98), item (0.79), work(verb, 0.78), quality(0.66)
Delivery	delivery (1.0), postage (1.0), ship (0.98), arrive (0.75), receive (0.46)
Communication	communication (1.0), response (1.0), email (1.0), reply (1.0)
Cost	cost (1.0), value (1.0), price (0.97), refund (0.71)
Transaction	ebayer (1.0), transaction (1.0), service (0.98), business (0.97), seller (0.96), deal (0.94), ebay (0.89), work (noun, 0.75), buy (0.74)

We apply the Stanford typed dependency relation parser to annotate words with Part of Speech (POS) tags and select nouns and verbs as candidates for dimension words. We further apply the association rule [Agrawal and Srikant, 1994] mining technique to determine *dimension words* – words that express a dimension with sufficient support ($\geq 1\%$) and confidence ($\geq 40\%$) are dimension words. Table 5.1 lists the dimension words together with their confidence – the conditional probability of a word being associated with a dimension [Agrawal and Srikant, 1994]. Note that words in the table are root words that represent various forms

[1] Annotation was done by two persons. Agreement is reached by discussions.

used in free text comments. For instance, *ship* also stands for *shipping* and *shipment*. Note also that when the word *work* is used as a verb, it refers to the Product dimension, as in the comment "*Works great and arrived before shipping estimated date*"; when *work* is used as a noun however, it refers to the overall transaction, as in the comment "*Good ebayer, great work*".

We automatically extract product information to augment identification of the Product dimension. In addition to general terms like *product* and *item*, specific product names are also frequently (approximately 15% of comments on the Product dimension) used in feedback comments. Fortunately most e-commerce systems keep meta-data on products that can be used to augment identification of the Product dimension. Each transaction on eBay is associated with a pre-defined hierarchy of product categories that can be obtained using the eBay API. We therefore identify the product dimension from feedbacks comments by including names at various levels of the corresponding product hierarchy. For example, in the comment "*My favourite lens. My favourite ebayer. These people are awesome.*", the lens is a camera part and at a layer in the product hierarchy for cameras on eBay. After extracting (*lens, favourite*) as a dependency relation pattern, we take *lens* to match names of the camera product hierarchy and as a result can identify that this comment expresses opinion on the product dimension. For Amazon feedback comments, as comments are not associated with product hierarchies for transactions, we extract keywords from the Amazon full store directory page and associate them with the Product dimension.

Table 5.2: Dimension-associated opinion expressions

Dimension	Eexpressions
Product	as described, beautiful, lovely
Delivery	on time, fast, lightning
Transaction	A+, 5 star, five star

Some opinion words are strongly associated with dimensions. For example, the word *beautiful* is strongly associated with the prodcut dimension whetheras words like *fast* and *lightning* are strongly associated with the delivery dimension. These dimension-associated opinion words are annotated in Table 5.2, and they are used to infer dimenions when the free text comments do not follow grammatical rules.

User feedback comments use informal language and often do not conform to any grammatical rules. Not surprisingly dimension ratings may not always be expressed as dimension-rating patterns that can be captured by the dependency relation parser. For example, the

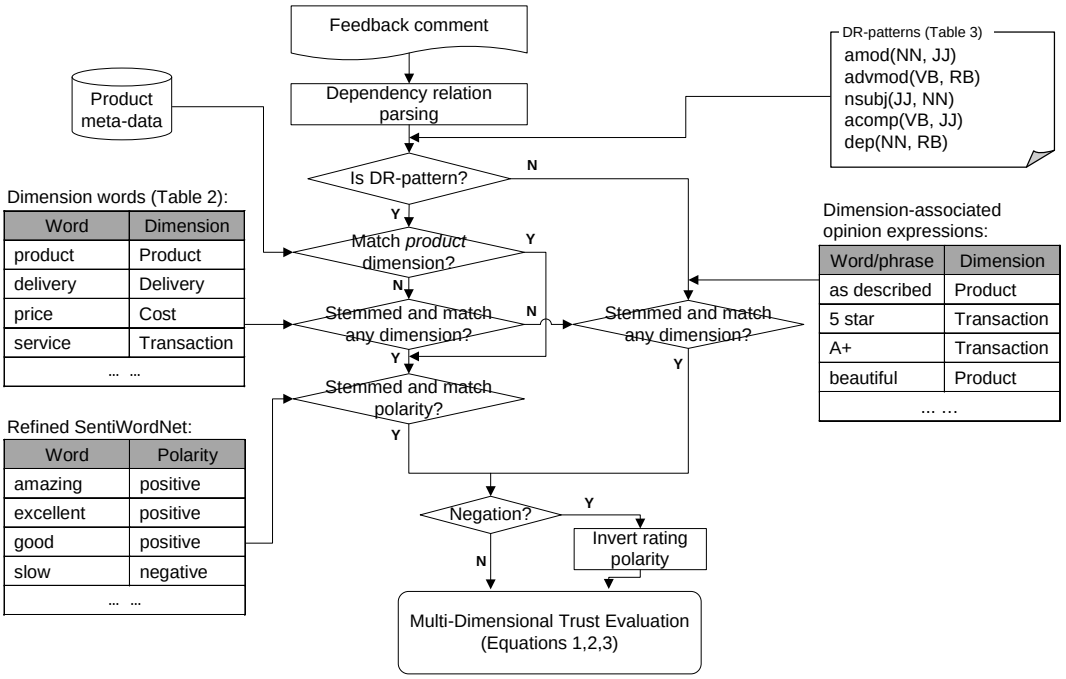


Figure 5.1: The CommTrust DR-mining algorithm

phrase "All 5 star" from the example comment in Figure 4.1 can not be identified as a dimension-rating pattern by the dependency relation parser. However the phrase indeed expressed opinion towards the Transaction dimension. In fact the phrase "All 5 star" is commonly used in feedback comments. A list of dimension-associated opinion phrases are compiled to help identify dimensions and associated ratings not expressed as dimension-rating patterns.

In addition to words, adjectival phrases are also used to express opinions. Similar to dimension-associated opinion words, they are dimension-associated phrases. $A+$ is parsed as a phrase by the Stanford parser and is a typical example. Although not containing any dimension word explicitly, $A+$ clearly expresses positive opinion towards buyers' overall experience – the transaction dimension. We include three commonly used opinion phrases with high conficence in our knowledge base – *as described* (97% confidence) expresses positive rating for the product dimension, and *5 star* (100 % confidence) and $A+$ (97% confidence) express positive rating for the transaction dimension. These dimension-associated opinion phrases are shown in Table 5.2.

The complete dimension-rating mining (DR-mining) algorithm for identifying dimensions and associated ratings from free text comments is shown in Figure 5.1. Each comment is first

analysed using the Stanford dependency relation parser. To identify dimensions in sentences, the dependency relations resulted from parsing are first matched against the DR-patterns shown in Table 4.1. If a DR-pattern is found, first the dimension is identified using the dimension word list in Table 5.1 as well as the associated product meta-data, and then the dimension rating is identified using the SentiWordNet opinion lexicon with refinement as described in Section 4.3.

When the dependency relation parsing result of a comment does not contain a DR-pattern, we try to discern if dimension ratings can be computed using dimension-associated opinion expressions, or if the comment contains at most three words, we try to identify if the comment expresses dimensional feedback by extracting the adjective and match it against dimension-associated opinion words (annotated words in Table 5.2).

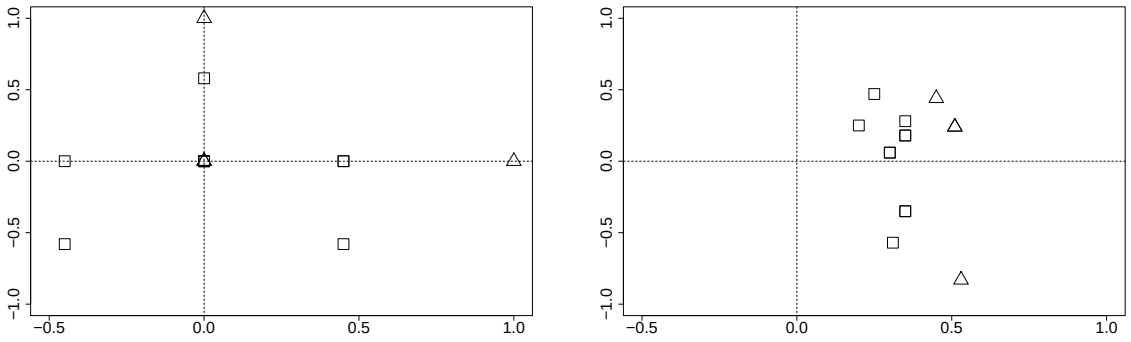
The negation relation (*neg()*) from the Stanford parser is used to detect any negation of ratings in DR-patterns. We go through the dependency relations for DR-patterns for the negation modifier *neg()* of the head or dependent. If either the head or dependent word of a DR-pattern involves the *neg()* relation, the relevant rating is inverted. The negation of feedback polarities expressed in dimension-associated words and phrases are identifyed using a negation word list [Goryachev et al., 2006].

In the process of matching dimension and opinion words, the Porter stemming algorithm [Porter, 1980] is used to achieve effective matching, considering the various formats for words.

Note that although the Transaction dimension is not considered in the trust model of CommTrust (refer to Equation 3.1), identifying ratings on the Transaction dimension is important for analysing feedback comments.

5.2 Computing dimension weights by matrix factorisation

Dimension trust scores need to be aggregated to compute the overall trust score for a seller. A simple approach is to average the dimension trust scores [Griffiths, 2005]. But the dimension ratings derived from feedback comments are highly noisy in two ways: many comments are from the same buyer and therefore are highly correlated; some buyers are lenient (or harsh) raters and therefore their ratings should be taken with a grain of salt. The problem we are facing is to compute dimension weights from noisy ratings. We propose to overcome the problem by assuming that there is some latent structure underlying the noisy ratings. Specifically we represent dimension ratings as a rating matrix and then compute the under-



(a) For the {-1, 0, +1} ratings

(b) For the {1, 3, 5} ratings

Figure 5.2: Latent component representations under SVD for the rating matrix in Table 3.2. Comment vectors and dimension vectors are represented as squares and triangles respectively.

lying structure for the ratings using singular value decomposition (SVD) [Deerwester et al., 1990]. Under SVD, comment vectors and dimension vectors are projected onto vectors in the same reduced space of latent independent components, where we can compute the weights for dimensions.

5.2.1 Singular value decomposition

With SVD, latent components in the new space are ordered so as to reflect major associative patterns in the original data, and ignore the smaller and less important influences. The full SVD of a matrix is defined as follows: Let $A_{m\times n}$ denote an $m \times n$ matrix, A can be decomposed into the product of three matrices:

$$A_{m\times n} = U_{m\times m} D_{m\times n} V^T_{n\times n}$$

where U contains the orthonormal eigen vectors for AA^T, V contains the orthonormal eigen vectors for A^TA, and D is a diagonal matrix containing the square roots of eigen values from U (V has the same eigen value as U). Especially D contains values indicating the variance of the original data points along each latent component, ordered in decreasing level of variance. The first component represents the largest variance of the original data points. In many applications (e.g. [Deerwester et al., 1990]), only the first several components of SVD (typically two or three) are considered to form a reduced representation as shown below,

where $k < m$ and $k < n$:

$$A_{m\times n} \approx U_{m\times k} D_{k\times k} V^T_{k\times n}$$

We take an example rating matrix from Table 3.2 in Chapter 3. The detailed results of the SVD for this example are shown in Appendix A and be ploted in Figure 5.2. With $k = 2$, the comment vectors and the dimension vectors are mapped to the same latent component space of two components. As shown in Figure 5.2(a), the first component is where original data points show most variance. In the latent component space, the row vectors of U are the five trust dimension vectors and the row vectors of V^T are the ten comment vectors. In Figure 5.2(a), squares represent comment vectors, and triangles denote dimension vectors. Note that in Figure 5.2(a) only three points are denoted for dimension vectors – the vectors for the Communication and Cost dimensions fall on the same position (0.00, 0.00), which indicates that Communication and Cost dimensions are highly correlated in the original rating matrix. Similarly only six points for comment rating vectors are denoted due to that four comment vectors fall on the same position, indicating that these comment ratings are highly correlated and possibly by the same seller.

As demonstrated in the example, representing dimensions in the latent component space has revealed the associative patterns in the original input data. More importantly such representation allows us to accurately compute the weights for trust dimensions, as discussed in the next section.

5.2.2 Computing dimension weights in the latent component space

With the ratings of {-1, 0, +1} for a rating matrix, the zeros are deemed missing values and the input matrix becomes very sparse. As a result, in addition to the high computation cost, SVD often results in latent vectors that are sparse and overlap in the latent space [Chandrasekaran et al., 2011]. Indeed in Figure 5.2(a) several points are at or close to the line with $x = 0$. Two comment rating vectors and two dimension rating vectors fall onto the point (0,0). To overcome this problem, and to accurately represent user ratings, we convert the user ratings of {-1, 0, +1} to a rating matrix of {1, 3, 5} and then SVD is applied. The detailed results see in Appendix A.

In Figure 5.2(b), we can see that points are further away from the line where $x = 0$. There are still some vectors fall onto the same point. For example, two dimension vectors fall onto the point (0.51, 0.24). With the reduced SVD $B_{4\times10} \approx U_{4\times2} D_{2\times2} V^T_{2\times10}$ (shown in Appendix A), the column vectors of $U_{4\times2}$ are the two components in the latent space

where ratings in $B_{4\times10}$ exhibit the largest variance and are ordered decreasingly by the level of variance. Specifically coefficients for the Product, Delivery, Communication and Cost dimensions for the first latent component are {0.53, 0.45, 0.51, 0.51}.

Note that column vectors in U are unit vectors. Specifically, for a column vector $\vec{u_j}$ of U ($j = 1..k$), the coefficients of $\vec{u_j}$ have the following property,

$$\sum_{d=1}^{m} \vec{u_j}[d]^2 = 1.$$

In our application, $d = 1..m$ ($m = 4$) indeed corresponds to the four dimensions Product, Delivery, Communication and Cost. The first column vector $\vec{u}_1$ represents the latent component where the original rating matrix demonstrates the highest variance and so potentially its coefficients can indicate weights for dimensions. However $\vec{u_1}$ itself may be unreliable as the weight for dimensions, while all k column vectors of U combined can provide a reliable estimation for dimension weights. Specifically the weight for a given dimension d is computed from k column vectors of U as follows:

$$w_d = \frac{\sum_{j=1}^{k} \vec{u_j}[d]^2}{k} \tag{5.1}$$

Typically we set $k = 2$ and our experiments confirm that this setting gives us reliable estimation of dimension weights while removing noises in the rating matrix. The dimension-weighted overall trust score for the original rating matrix can be computed by aggregating the dimension trust scores computed from Equation 3.3 with the weights computed from Equation 5.1.

With the above SVD of rating matrix B with $k = 2$, the weights for dimensions Product, Delivery, Communication and Cost are computed from the two column vectors of U, as shown below:

$$\vec{w}^T = \begin{bmatrix} 0.48 & 0.20 & 0.16 & 0.16 \end{bmatrix}$$

It can be seen that Communication and Cost have the lowest weights, and Product has the highest weight, which intuitively corresponds to the total number of positive and negative ratings and their level of consistency in ratings along these dimensions. As shown in the rating matrix A, the Product dimension has a small number of positive and negative ratings, and the highest variation – three positive, two negative and five neutral ratings. The Communication and Cost dimensions are dominated by neutral ratings and have the highest

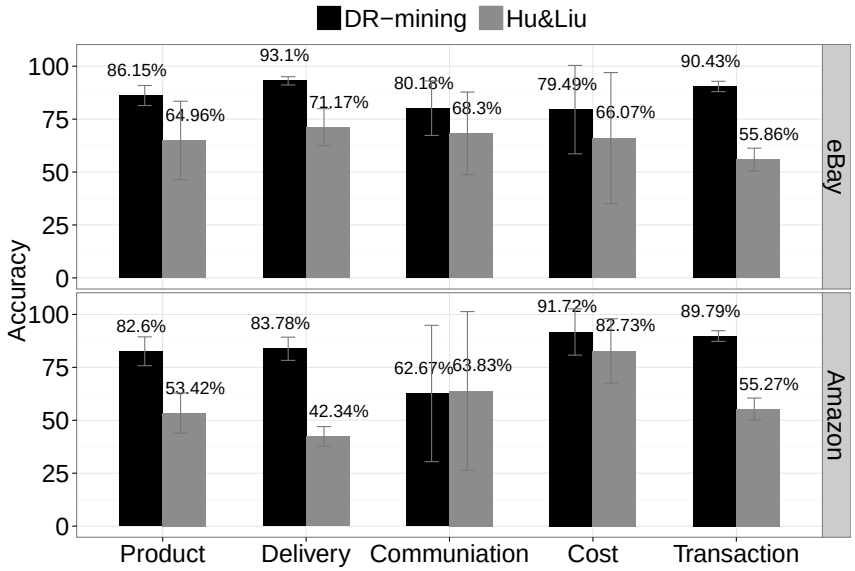


Figure 5.3: Dimension accuracy of the CommTrust DR-mining algorithm on eBay and Amazon data

consistence in ratings – one positive and nine neutral ratings.

Given the weights as computed above in $\vec{w}^T$, and the dimension trust scores in Example 3.1.1, according to Equation 3.1 the overall trust score for the seller in Table 3.2 is 53.4%. On the other hand, based on direct ratings on transactions (the "Tran." column of Table 3.2), the transaction trust score for the seller computed from Equation 3.3 is 76.92%. In contrast, based on the overall ratings recorded on eBay (the "eBay" column of Table 3.2), the positive rating percentage score is 8/9 or 88.9%. It can be seen that the eBay reputation system, not considering dimension trusts and not considering the neutral ratings, the high overall trust score is very likely inflated – given that ratings for Communication and Cost dimensions are as low as only one positive rating out of ten comments.

5.3 Experiments

5.3.1 Accuracy of the CommTrust DR-mining algorithm

The most prominent retrieval measures are precision and recall [Baeza-Yates et al., 1999]. Recall is defined as the ratio between the number of retrieved relevant items to the total number of existing relevant items. Precision is defined as the ratio between the number of

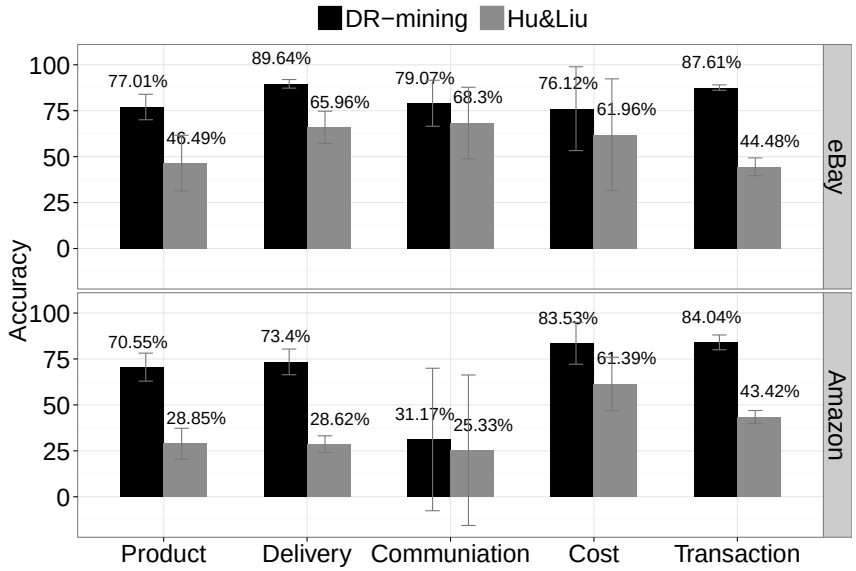


Figure 5.4: Dimension rating accuracy of the CommTrust DR-mining algorithm on eBay and Amazon data

relevant items and the total of trtrieved items. The goal is to maximise both, but commonly they have antogonisitc behaviour, i.e., trying to increase recall will likely reduce precision. To compare different systems, combinations of precision P and recall R metrics have been developed, such as the F_1 measure, $F_1 = 2PR/(P + R)$, which can also be generalised to a weighted F_1 measure, $F_w = (\lambda_P + \lambda_R)PR/(\lambda_P P + \lambda_R R)$. With the given weightings, the preferences to precision or recall can be adjusted. A direct relation between precision and recall to perplexity and language models has been given in [Azzopardi et al., 2003]. We apply recall to evaluate accuracy of DR-mining algorithm.

1,956 feedback comments between 31 January and 18 March 2012 for ten sellers on eBay are used to build the dimension lexicon for the CommTrust DR-mining algorithm. For cross validation, we evaluate the performance of CommTrust, the trust model as well as DR-mining algorithm, on ten other random sellers on eBay and ten random sellers on Amazon. 200 comments for each seller are randomly selected, and non-English comments are removed. Each comment is annotated with dimensions and their associated ratings of positive (+1), negative (-1) and neutral (0) [2].

To evaluate the CommTrust DR-mining algorithm, as a baseline we implemented the

[2]Annotation is done by two persons. Agreement is reached by discussions.

Table 5.3: Overall trust scores and ranks for ten eBay sellers and ten Amazon sellers

eBay					Amazon				
seller	CommTrust		eBay		seller	CommTrust		Amazon	
	trust	rank	trust	rank		trust	rank	trust	rank
EA	0.88	7	1.00	1	AA	0.86	4	0.98	1
EB	0.90	3	1.00	2	AB	0.87	3	0.97	4
EC	0.89	5	0.99	7	AC	0.89	1	0.98	2
ED	0.89	6	0.99	8	AD	0.82	10	0.97	5
EE	0.84	9	0.98	9	AE	0.84	8	0.96	8
EF	0.92	1	1.00	3	AF	0.86	5	0.97	6
EG	0.90	4	1.00	4	AG	0.88	2	0.94	9
EH	0.84	10	1.00	5	AH	0.85	6	0.93	10
EI	0.91	2	1.00	6	AI	0.85	7	0.98	3
EJ	0.87	8	0.98	10	AJ	0.84	9	0.97	7
Kendall's τ				0.3333					0.1556
p-value				0.2164					0.6007

Hu&Liu algorithm as described in [Hu and Liu, 2004b], as follows:

- The Stanford Part of Speech Parser is first used for POS tagging. From the 1,956 feedback comments for eBay sellers used for training, nouns with a frequency of $\geq$ 1% are deemed dimension words, and are then assigned to the four dimensions and Transaction.
- Adjectives and adverbs close (by default three words before and after) to the dimension words are opinion words. SentiWordNet is used to decide the polarity of opinion words.
- Negation words are based on the negation word list in [Goryachev et al., 2006], and are applied as within two words before opinion words.

Figure 5.3 shows the average accuracies of dimension identification for the CommTrust DR-mining algorithm and the Hu&Liu approach on feedback comments of ten eBay sellers and ten Amazon sellers for evaluation. Note that the accuracies for four dimensions used in the trust model of CommTrust as well as that for the Transaction dimension are included. On both datasets, CommTrust significantly outperforms the Hu&Liu approach for identifying all dimensions(Wilcoxin signed rank test, $p < 0.05$). On the eBay dataset CommTrust achieves accuracies from 79.49% on the Cost dimension to 93.1% on the Delivery dimension, whereas Hu&Liu can only reach 71.17% on the Delivery dimension. On the Amazon dataset, the accuracy of CommTrust varies from 62.67% for the Communication dimension to 91.72%

Table 5.4: Dimension trust scores for ten eBay sellers

Dimension		eBay Sellers									
		EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ
Product	CT	0.91	0.90	0.93	0.94	0.91	0.95	0.93	0.96	0.90	0.90
	eBay	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.96
Delivery	CT	0.95	0.97	0.95	0.92	0.88	0.96	0.95	0.83	0.94	0.96
	eBay	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.94
Comm	CT	0.73	0.75	0.62	0.77	0.73	0.70	0.75	0.75	0.85	0.71
	eBay	0.98	0.98	0.94	0.96	0.94	0.98	0.96	0.96	0.98	0.94
Cost	CT	0.76	0.83	0.82	0.79	0.57	0.86	0.75	0.57	0.86	0.56
	eBay	1	1	0.98	1	0.98	1	1	1	1	0.98

CT: CommTrust

Table 5.5: Dimension trust scores for ten Amazon sellers

Dimension	Amazon Sellers									
	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ
Product	0.89	0.89	0.93	0.77	0.87	0.89	0.95	0.91	0.88	0.89
Delivery	0.93	0.96	0.95	0.92	0.93	0.94	0.91	0.93	0.94	0.89
Comm	0.57	0.57	0.57	0.62	0.57	0.57	0.50	0.50	0.50	0.62
Cost	0.75	0.78	0.75	0.81	0.73	0.73	0.81	0.67	0.75	0.76

CT: CommTrust

for the Cost dimension, whereas the Hu&Liu accuracy varies from 42.34% on the Delivery dimension to 63.83% on the Communication dimension. The biggest gap in accuracy between CommTrust and Hu&Liu lies in identifying the Transaction dimension. This is mainly due to that phrases are often used to express ratings directly, without explicitly using any nouns for transactions. In CommTrust the Transaction dimension can be inferred from the dimension-associated opinion expressions but they are missed by the Hu&Liu approach.

Figure 5.4 compares the accuracies of dimension rating identification of CommTrust and Hu&Liu, averaged over ten sellers eBay and Amazon respectively. Identifying dimension ratings depends on identifying dimensions first, and so the accuracy for dimension rating identification is generally lower than that for identifying the corresponding dimension. CommTrust generally achieves consistently high accuracy. On the eBay dataset, it has accuracies of 76.12% for Cost rating to 89.64% for Delivery rating. On the Amazon dataset, it achieves accuracies from 70.55% for Product rating to 84.04% for Transaction rating. In contrast, Hu&Liu achieves reasonably good accuracy on the eBay dataset – from 44.48% for Transaction rating to 68.3% for Communication rating, but very modest accuracy on the

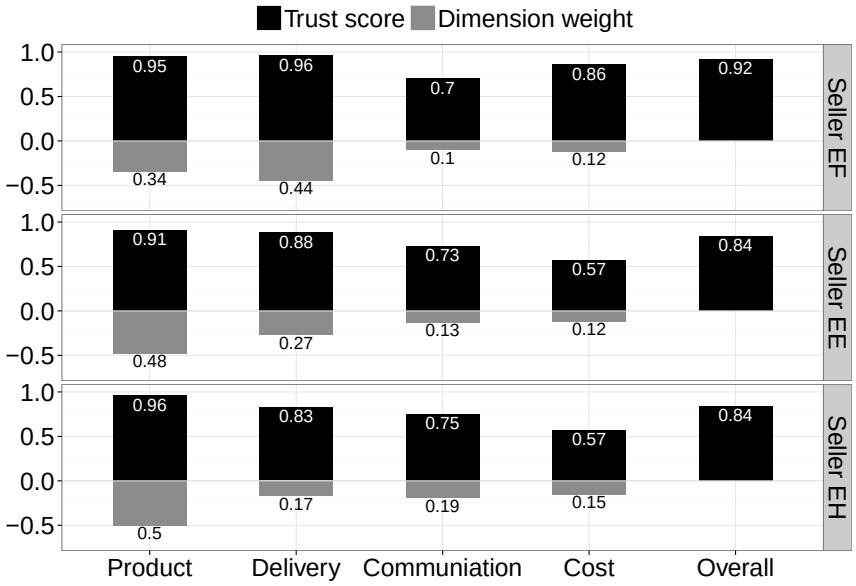


Figure 5.5: The comprehensive trust profiles by CommTrust for eBay sellers

Amazon dataset – from 25.33% for Communication rating to 61.39% for Cost rating. The consistently higher accuracy of CommTrust is due in part to the application of dependency relation analysis for DR-patterns to identify opinion words and in part to the application of a refined opinion lexicon of SentiWordNet. The odd low accuracy with high variance for both CommTrust and Hu&Liu on Communication ratings on the Amazon data results from the fact that comments on Communication is extremely rare – four out of ten sellers have only one comment on Communication and missing this only rating results in an accuracy of zero.

5.3.2 Evaluation of the trust model

Table 5.3 shows the overall trust scores and rankings for ten eBay sellers and ten Amazon sellers, where dimension weights are computed using Equation 5.1 ($k = 2$). It can be seen from Table 5.3 that the eBay feedback score is generally very high (0.98–1.00) for all ten sellers, and it is impossible to rank sellers. With 5-star ratings of finer grain, the Amazon feedback score partially remedy the problem, but still the feedback scores are high and very close to each other (0.93–0.98). The CommTrust overall trust scores are at a more reasonable level, and more importantly can effectively rank sellers. The ranking by CommTrust does not correlate with the eBay system ($\tau = 0.3333$, $p = 0.2164$) for eBay sellers, it also does

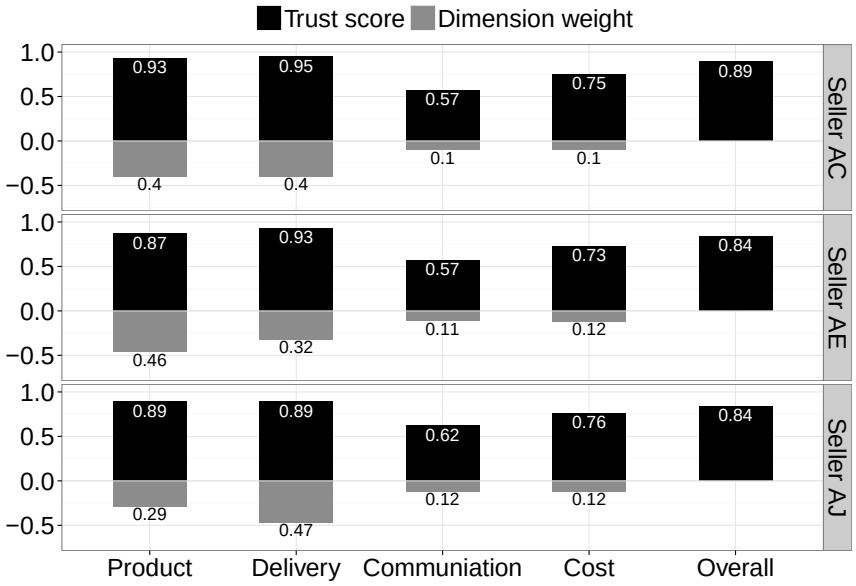


Figure 5.6: The comprehensive trust profiles by CommTrust for Amazon sellers

not correlate to the Amazon system ($\tau = 0.1556$, $p = 0.6007$) for Amazon sellers. This is suggesting that CommTrust has provided distinct information that is not included in any other models.

The dimension trust scores by CommTrust are shown in Table 5.4 and Table 5.5. The dimension trust scores by CommTrust are at a more reasonable level and sellers show different trust profiles along different dimensions. On the Cost dimension for example, the trust scores by CommTrust for eBay sellers vary from 0.56 for seller EJ to 0.86 for seller EI. Note that the trust scores for Communication and Cost are much lower than those for Product and Delivery. This is due to the lack of ratings on Communication and Cost in comments. Nevertheless CommTrust can effectively rank sellers on dimensions. The rankings of sellers on dimensions do not correlate (max $\tau = -0.32$, with $p = 0.26$). This is showing that the CommTrust dimension trust model can effectively capture the differences between dimensions. Table 5.4 also lists the DSR scores by the eBay reputation system for eBay sellers on the corresponding dimensions. Obviously all ten eBay sellers have consistently high DSR scores (0.94–1). Moreover, ranking by DSR scores on Product is highly correlated with that on Delivery (Kendall's $\tau = 1$, $p = 0.01$) and ranking on Communication is highly correlated with that on Cost ($\tau = 0.80$, $p = 0.02$). It is obviously difficult to rank sellers dimensions by the eBay DSR scores.

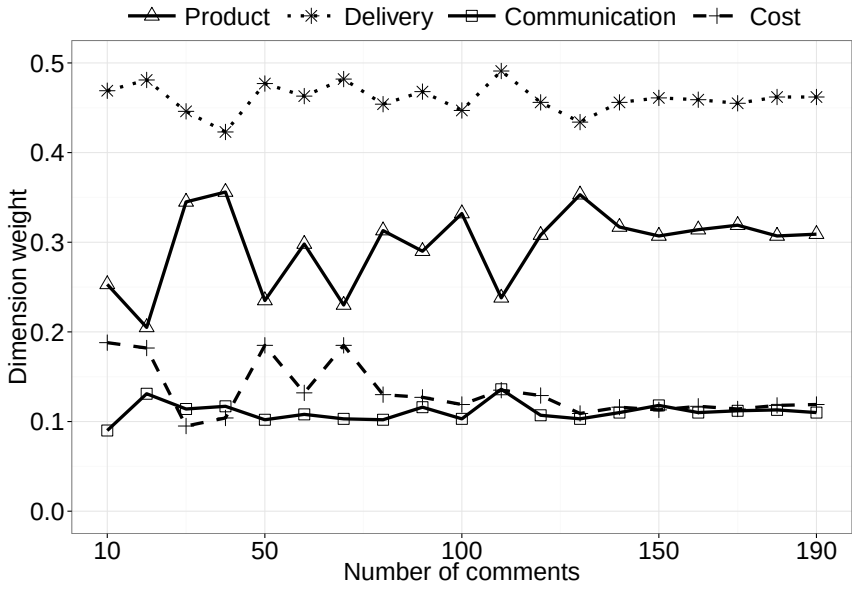


Figure 5.7: Dimension trust weight with respect to number of comments on eBay

5.3.3 Trust profiles for sellers

The dimension trust scores and weights together form the complete trust profiles for sellers in e-commerce applications. Figure 5.5 and Figure 5.6 depicts the different trust profiles for three representative eBay and Amazon sellers respectively. For each seller, the upward bars represent trust scores for dimensions while the downward bars represent their weights. For example in Figure 5.6, while having the highest overall trust score of 0.89, Amazon seller AC has a low dimension trust score of 0.57 for Communication. Sellers AE and AJ have the same overall trust score of 0.84, but their dimension trust profiles are different. Seller AE has a high trust score for Delivery with a low weight of 0.32 whereas seller AJ has a lower trust score for Delivery with a high weight of 0.47. The complete trust profiles of eBay sellers and Amazon sellers as shown in Figure 5.5 and Figure 5.6 clearly illustrate the variation of dimension trust for each seller horizontally and those across different sellers vertically. Such comprehensive trust profiles certainly can cater to buyers' preferences for different dimensions and guide buyers in making informed decisions when choosing sellers.

In Figure 5.7 and Figure 5.8, the dimension weights for eBay seller EB and Amazon seller AE are plotted, with respect to the number of comments (Dimension weights for other sellers show similar trends and are omitted to save space). In the figure, when the number of comments is increased from ten to 190 (recall that a seller has 200 comments) the dimen-

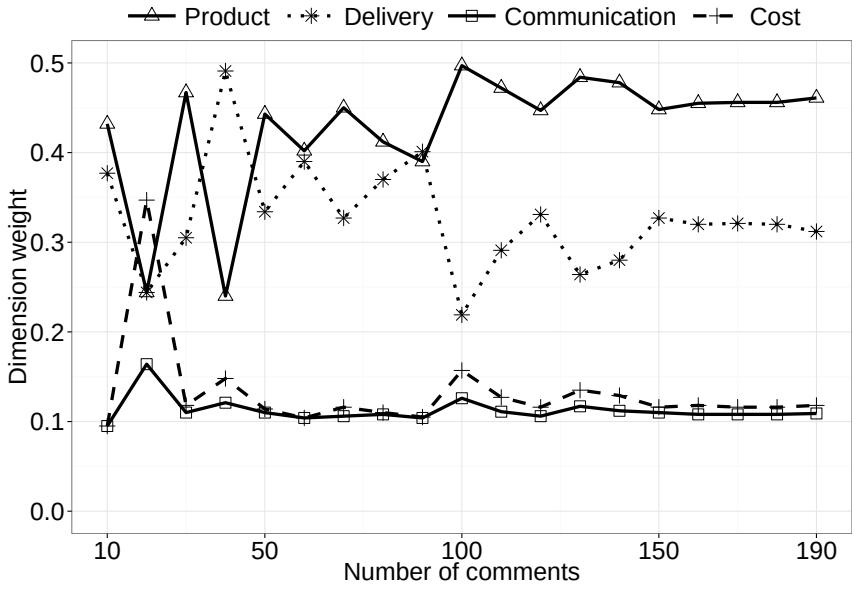


Figure 5.8: Dimension trust weight with respect to number of comments on Amazon

sion weights converge at 140 and 150 comments for eBay seller EB and Amazon seller AE respectively. This suggests that a reasonable number of comments is sufficient to reliably estimate dimension weights. It is obvious from the figure that Communication and Cost dimensions have low weights (0.11 and 0.12 respectively), whereas Product and Delivery have high weights (0.31 and 0.46 respectively). This result is consistent with our observation that there are few positive and negative ratings on Communication and Cost in feedback comments.

5.4 Summary

In this chapter, we have answered research questions on how to identify dimension from feedback comments and how to evaluate the weights of each dimension.

We have proposed a knowledge-based approach that incorporates domain knowledge, meta-data, and general grammatical patterns to accurately identifying dimension ratings from feedback comments.

We have formulated the problem of computing dimension weights from ratings as a factor analytic problem and propose a matrix factorisation technique to automatically compute weights for dimensions from the sparse and noisy dimension rating matrix.

The findings from the experiments on evaluating the overall trust score illustrate the

rankings is correlated to the results from the previous user study that we conducted, as explained in Chapter 3. This shows that our proposed trust model is efficient to rank sellers and identify a trustworthy seller. Moreover, the accuracy of DR-mining algorithm is conducted to evaluate the dimension accuracy and the dimension rating accuracy on eBay and Amazon datasets. This shows our algorithm can achieve reasonable results on identifying dimension and dimension ratings.

Chapter 6

Conclusions and Future work

The "*all good sellers*" problem is well known for the reputation management systems of popular e-commerce web sites like eBay and Amazon. The high reputation scores for sellers can not effectively rank sellers and therefore can not guide potential buyers to select trustworthy sellers to transact with. On the other hand, it is observed that although buyers may give high feedback ratings on transactions, they often express direct negative opinions on aspects of transactions in free text feedback comments.

We have proposed a multi-dimensional trust evaluation model CommTrust for computing comprehensive trust profiles for sellers in e-commerce applications. Different from existing multi-dimensional trust models, we compute dimension trust scores and dimension weights automatically via extracting dimension ratings from feedback comments. Our experiments on eBay and Amazon data show that CommTrust can compute comprehensive trust profiles for sellers that manifest distinct valuable information not available in feedback scores by eBay or Amazon reputation systems. Our model can distinctively identify the reputable sellers from another seller that have had bad history with previous buyers. Moreover, the ratings are more reasonable and acceptable, and not all sellers have high scores, as compared to other e-commerce websites. It can significantly reduce the strong positive bias in e-Commerce reputation systems, and solve the "all good sellers" problem. This model is good assistance to the buyers when doing online transaction, as to shield them from being a victim of fraud and untrusted sellers.

The research questions presented in the introduction relate to building CommTrust, a comprehensive trust model. This model is aimed as a guideline for potential buyers to assess the most trusted and reliable sellers. In the following we will summarize how each research

question are answered.

Research question 1: How can multi-dimensional trust from extracted dimensions and the associated opinion polarity be computed?

In e-commerce environments, different transactions may have different contexts. The trustworthiness of a seller displayed on the e-commerce websites should be a reliable indication for the potential buyers to purchase items or service from them. Therefore, our trust model is aimed to provide a clear and objective ranking for the sellers. The trust model, which is built from the previous feedback comments would assist the potential buyers in making the right decision when attempting to purchase from a particular seller.

Research question 2: How can dimensions from online feedback comments that customers have expressed their opinions on be more accurately identified?

The first approach applied topic modelling to automatically generate the dimensions. We proposed the Lexical-LDA algorithm to group aspect expressions into semantically coherent categories, which we refer as dimensions. These dimensions are based on lexical knowledge which discussed in Section 4.2. This approach is able to achieve more effective clustering, can be used across different domains and also to avoid manual time-consuming annotation. The findings of the experiment on the e-commerce data set using Lexical-LDA illustrate that the Rand Index is over 0.75 and the accuracy is around 70%. The dimensions are more accurate and easy to understand. We also applied this technique to another data set, which is a hotel review data set. The results showed stable performance across domains. In addition to that, the features are more transparent and easy to understand by users.

In the second approach, given the five dimensions based on eBay Seller Detailed Ratings, which are products as described, communication, postage time and charges, we proposed a Dimension Rating mining algorithm to automatically identify the dimension rating expressions from the comments such as “nice item”, “quick shipping”, “great service” and many more. The five dimensions that we used in our study are products, delivery, communication, cost and transaction (the definition of these five dimensions presented in Section 5.1). From the training data, we extracted these five dimension word list and the dimension-associated opinion expressions. We also used the product

metadata from eBay and Amazon to identify the product dimension. The experiment shows a comparatively better baseline from Hu&Liu Hu and Liu [2004b] approach.

Research question 3: How can weights for each dimension that customers have expressed their preference on be more efficiently evaluated?

We applied latent factor models to discover the latent features from the observed data, which is the dimension rating matrix.

In lexical-LDA approach, automatically group the aspect opinion expressions. The total number of expressions is normalized to produce the dimension weight.

In matrix factorization model, we observed the occurrences of terms in the dimensions and assign rating to the respective dimensions. From here, we applied Singular Value Decomposition (SVD) to estimate the uncorrelated factors. These factors are then used to compute the weight for each dimension.

These weights are compared with the user study ranking results. The findings show that applying the weights, the ranking has significant correlation with user study results. In contrast, the same model without these weights, the ranking is not correlated to the user study results. Thus, this proves that the weights computed using latent factor is accurate in determining the aspect opinion expressions.

Research question 4: How can sentiment from textual feedback for each dimension be more accurately classified?

Sentiment classification aims to identify view-point from information expressed in text. Whether a piece of text is expressing positive or negative attitude towards associated aspect of comments need to be identified.

We applied general opinion word lexicon SentiWordNet, which is a widely used public domain NLP resource to identify opinion polarities. SentiWordNet has three different scores; positive, negative and neutral. Since our research only requires positive or negative score, we add the sum of positive and negative scores. If it is greater than or equal to 0.5, then only it is considered as an aspect opinion expressions. If the positive score is higher, it carries the positive polarity, and if negative score is higher, it carries the negative polarity. Our experiments on both approaches, topic modelling and machine learning approachs, showed better than baseline.

Future work

The CommTrust proposed in this thesis can be used to reliably evaluate the trustworthiness of sellers. However, it is still need improvement to mine more detailed information from feedback comments.

In on-line feedback comments, casual language is commly used to express users' opinon. For example, some users type in "prod" to refer as "product". Currectly in our research, when identifying the term in comments, we relied on the type dependency relations, ignored the spelling. As the results of dimension terms, "prod" and "product" may both identified. In future work, we can improve mining techniques to identify terms more acurately.

Currently, we applied the SentiWordNet to distinguish the positive and negative comments. Future work can explore the possibility of understanding the contents more in-depth. For example, a comment that states the products price is cheap does not necessarily means the product is not of a good quality. Further more, the CommTrust only take into consideration of the positive and negative comments. Future work also can include the neutral opinions in the comments as the input to build the trust model.

In e-commerce reputation systems, users can leave text comments and an overall ranking score based on their experience. In order to solve the "all good repuation" problem, in our research we only look at the comments regarless the overall ranking score. Somehow the overall ranking rated by the user is useful information on some level. Future work can be expanded by including the overall ranking from the users with the comments and compute a new trust value.

Appendix A

An SVD Example

This appendix presents an SVD example using the sample data of dimension ratings for a seller on eBay.

The 4×10 matrix A below represents a rating matrix. The row vectors of A are the dimension vectors, where each of ten components corresponds to a comment. The column vectors of A are the comment vectors, where each of four components corresponds to a dimension.

$$A = \begin{array}{c} \begin{array}{cccccccccc} c_1 & c_2 & c_3 & c_4 & c_5 & c_6 & c_7 & c_8 & c_9 & c_{10} \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Applying the reduced SVD model with $k = 2$ to matrix A produces the decomposition as

follows:

$$A \approx UDV^T = \begin{bmatrix} 1.00 & 0.00 \\ 0.00 & 1.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \end{bmatrix} \begin{bmatrix} 2.24 & 0 \\ 0 & 1.73 \end{bmatrix} \begin{bmatrix} 0.45 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.45 & -0.58 \\ 0.00 & 0.00 \\ 0.00 & 0.00 \\ 0.00 & 0.58 \\ 0.45 & 0.00 \\ -0.45 & -0.58 \\ -0.45 & 0.00 \end{bmatrix}^T$$

The above rating matrix A on ratings {-1, 0, +1} is mapped to a rating matrix B on ratings {1, 3 5} as follows:

$$B = \begin{bmatrix} 5 & 3 & 3 & 5 & 3 & 3 & 3 & 5 & 1 & 1 \\ 3 & 3 & 3 & 1 & 3 & 3 & 5 & 3 & 1 & 3 \\ 3 & 3 & 5 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 3 & 3 & 3 & 5 & 3 & 3 & 3 & 3 \end{bmatrix}$$

SVD produces the following reduced decomposition for B:

$$B \approx UDV^T = \begin{bmatrix} 0.53 & -0.83 \\ 0.45 & 0.44 \\ 0.51 & 0.24 \\ 0.51 & 0.24 \end{bmatrix} \begin{bmatrix} 19.88 & 0 \\ 0 & 4.06 \end{bmatrix} \begin{bmatrix} 0.35 & -0.35 \\ 0.30 & 0.06 \\ 0.35 & 0.18 \\ 0.31 & -0.57 \\ 0.30 & 0.06 \\ 0.35 & 0.18 \\ 0.35 & 0.28 \\ 0.35 & -0.35 \\ 0.20 & 0.25 \\ 0.25 & 0.47 \end{bmatrix}^T$$

Bibliography

R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. Int. Conf. on VLDB*, 1994.

L. Azzopardi, M. Girolami, and K. van Risjbergen. Investigating the relationship between language model perplexity and ir precision-recall measures. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 369–370. ACM, 2003.

S. Baccianella, A. Esuli, and F. Sebastiani. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proc. 7th Int. Conf. on Language Resources and Evaluation*, 2010.

R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

J. Blitzer, M. Dredze, and F. Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, volume 7, pages 440–447, 2007.

S. Brody and N. Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Proc. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812, 2010.

G. Carenini, R. Ng, and E. Zwart. Extracting knowledge from evaluative text. In *Proceedings of the 3rd international conference on Knowledge capture*, pages 11–18. ACM, 2005.

G. Casella and R. L. Berger. *Statistical inference*. Duxbury Press, 1990.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

M. De Marneffe and C. Manning. The stanford typed dependencies representation. In *Proc. the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, 2008.

M. De Marneffe, B. MacCartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proc. LREC*, volume 6, pages 449–454, 2006.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

A. Esuli and F. Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422, 2006.

A. Fahrni and M. Klenner. Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proc. of the Symposium on Affective Language in Human and Machine, AISB*, pages 60–63, 2008.

J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378, 1971.

M. Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *20th Int. Conf. on Computational Linguistics*, 2004.

S. Goryachev, M. Sordo, Q. Zeng, and L. Ngo. Implementation and evaluation of four different methods of negation detection. Technical report, Technical report, DSG, 2006.

N. Griffiths. Task delegation using experience-based multi-dimensional trust. In *Proc. the fourth international joint conference on AAMAS*, pages 489–496, 2005.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

G. Heinrich. Parameter estimation for text analysis. Technical report, University of Leipzig, 2005.

Y. Hijikata, H. Ohno, Y. Kusumura, and S. Nishida. Social summarization of text feedback for online auctions and interactive presentation of the summary. *Knowledge-Based Systems*, 20(6):527–541, 2007.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, 2004.

M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proc. the National Conference on Artificial Intelligence*, pages 755–760, 2004a.

M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proc. 4th Int. Conf. on KDD*, pages 168–177, 2004b.

A. Jøsang and R. Ismail. The beta reputation system. In *Proc. the 15th bled electronic commerce conference*, pages 41–55, 2002.

A. Jøsang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decision Support Systems*, 43(2):618–644, 2007.

S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. the 12th Int. Conf. on WWW*, 2003.

K. Karplus. Evaluating regularizers for estimating distributions of amino acids. In *Proc. Third Int. Conf. on Intelligent Systems for Molecular Biology*, volume 3, pages 188–196, 1995.

S. Kim and E. Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics, 2004.

Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. 14th Int. Conf. on KDD*, pages 426–434, 2008.

Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53 (4):89–97, 2010.

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

S. Kubler, R. McDonald, J. Nivre, and G. Hirst. *Dependency Parsing*. Morgan and Claypool Publishers, 2009.

J. R. Landis and G. G. Koch. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174, 1977.

C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proc. the 18th ACM conference on Information and knowledge management*, pages 375–384, 2009.

B. Liu. *Sentiment analysis and opinion mining*. Morgan & Claypool Publishers, 2012.

Y. Lu, C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *18th Int. Conf. on WWW*, 2009.

C. Manning and H. Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

A. K. McCallum. MALLET: a machine learning for language toolkit, 2002. URL http://mallet.cs.umass.edu.

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proc. the 16th international conference on World Wide Web*, pages 171–180, 2007.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to wordnet: An on-line lexical database*. *International journal of lexicography*, 3(4):235–244, 1990.

A. Mukherjee and B. Liu. Aspect extraction through semi-supervised modeling. In *Proc. the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 339–348, 2012.

J. ODonovan, B. Smyth, V. Evrim, and D. McLeod. Extracting and visualizing trust relationships from online auction feedback comments. In *Proc. IJCAI'07*, pages 2826–2831, 2007.

B. Ohana and B. Tierney. Sentiment classification of reviews using SentiWordNet. *9th. IT & T Conference*, 2009.

B. Pang and L. Lee. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2 (1-2):1–135, Jan. 2008.

A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup and Workshop*, 2007.

A. Popescu and O. Etzioni. Extracting product features and opinions from reviews. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 339–346. Association for Computational Linguistics, 2005.

M. F. Porter. An algorithm for suffix stripping, 1980.

G. Qiu, B. Liu, J. Bu, and C. Chen. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international jont conference on Artifical intelligence*, pages 1199–1204. Morgan Kaufmann Publishers Inc., 2009.

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

S. Ramchurn, D. Huynh, and N. Jennings. Trust in multi-agent systems. *The Knowledge Engineering Review*, 2004.

W. M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

S. Reece, A. Rogers, S. Roberts, and N. Jennings. Rumours and reputation: Evaluating multi-dimensional trust within a decentralised reputation system. In *Proc. the 6th international joint conference on AAMAS*, pages 165–172, 2007.

P. Resnick and R. Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *The Economics of the Internet and E-commerce*, 2002.

P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation Systems: Facilitating Trust in Internet Interactions. *Communications of the ACM*, 43:45–48, 2000.

P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The value of reputation on ebay: A controlled experiment. *Experimental Economics*, 9(2):79–101, 2006.

A. Rettinger, M. Nickles, and V. Tresp. Statistical relational learning of trust. *Machine learning*, 82:191–209, 2011.

J. Sabater and C. Sierra. Regret: reputation in gregarious societies. In *Proc. the fifth international conference on Autonomous agents*, pages 194–195. ACM, 2001.

F. P. . R. M. Schillo, M. Using trust for detecting deceptive agents in artificial societies. *Applied Artificial Intelligence*, 14(8):825–848, 2000.

D. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. CRC PressI Llc, 2004.

B. Shi and K. Chang. Mining chinese reviews. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 585–589. IEEE, 2006.

G. Somprasertsri and P. Lalitrojwong. Extracting product features and opinions from product reviews using dependency analysis. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2010 Seventh International Conference on*, volume 5, pages 2358–2362. IEEE, 2010.

M. Stark and R. Riesenfeld. Wordnet: An electronic lexical database. In *Proceedings of 11th Eurographics Workshop on Rendering*, 1998.

G. Takàcs, I. Pilàszy, B. Németh, and D. Tikk. Major components of the gravity recommendation system. *ACM SIGKDD Explorations Newsletter*, 9:80–84, 2007.

D. R. Thomas. A general inductive approach for analyzing qualitative evaluation data. *American journal of evaluation*, 27(2):237–246, 2006.

P. Thomas and D. Hawking. Evaluation by comparing result sets in context. In *Proc. the 15th ACM international conference on Information and knowledge management*, pages 94–101, 2006.

I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proc. the 17th international conference on World Wide Web*, pages 111–120, 2008a.

I. Titov and R. T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proc. ACL*, pages 308–316, 2008b.

P. D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proc. the 40th annual meeting on association for computational linguistics*, pages 417–424, 2002.

H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proc. the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792, 2010.

H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis without aspect keyword supervision. In *Proc. the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 618–626, 2011.

X. Wang, L. Liu, and J. Su. RLM: A general model for trust representation and aggregation. *IEEE Transactions on Services Computing*, 5(1):131–143, 2012.

Y. Wang and E. Lim. The evaluation of situational transaction trust in e-service environments. In *IEEE International Conference on e-Business Engineering*, pages 265–272, 2008.

Y. Wang and M. Singh. Trust representation and aggregation in a distributed agent system. In *Proc. the national conference on Artificial Intelligence*, 2006.

L. Xiong and L. Liu. A reputation-based trust model for peer-to-peer e-commerce communities. In *Proc. IEEE Int. Conf. E-Commerce*, pages 275–284, 2003.

L. Xiong and L. Liu. Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

B. Yu and M. P. Singh. Distributed reputation management for electronic commerce. *Computational Intelligence*, 2002.

G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.

Z. Zhai, B. Liu, H. Xu, and P. Jia. Constrained LDA for grouping product features in opinion mining. In *Proc. Advances in knowledge discovery and data mining*, pages 448–459. Springer, 2011.

H. Zhang, Y. Wang, and X. Zhang. Efficient contextual transaction trust computation in e-commerce environments. In *Proc. 11th IEEE Int. Conf. on Trust, Security and Privacy in Computing and Communicatios (TrustCom 2012)*, 2012a.

H. Zhang, Y. Wang, and X. Zhang. A trust vector approach to transaction context-aware trust evaluation in e-commerce and e-service environments. In *Proc. 5th IEEE Int. Conf. on Service-oriented Computing and Applications (SOCA 2012)*, 2012b.

Y. Zhang and Y. Fang. A fine-grained reputation system for reliable service selection in peer-to-peer networks. *IEEE Transactions on Parallel and Distributed Systems*, 2007.

W. X. Zhao, J. Jiang, H. Yan, and X. Li. Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid. In *Proc. the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 56–65, 2010.

L. Zhuang, F. Jing, X. Zhu, and L. Zhang. Movie review mining and summarization. In *Proc. 15th ACM Int. Conf. on Information and knowledge management*, pages 43–50, 2006.