OPINION INTEGRATION AND SUMMARIZATION

BY

YUE LU

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2011

Urbana, Illinois

Doctoral Committee:

Associate Professor ChengXiang Zhai, Chair & Director of Research
Professor Jiawei Han
Professor Dan Roth
Assistant Professor Panayiotis Tsaparas, University of Ioannina

Abstract

As Web 2.0 applications become increasingly popular, more and more people express their opinions on the Web in various ways in real time. Such wide coverage of topics and abundance of users make the Web an extremely valuable source for mining people's opinions about all kinds of topics. However, since the opinions are usually expressed as unstructured text scattered in different sources, it is still difficult for the users to digest all opinions relevant to a specific topic with the current technologies. This thesis focuses on the problem of opinion integration and summarization whose goal is to better support user digestion of huge amounts of opinions for an arbitrary topic. To systematically study this problem, we have identified three important dimensions of opinion analysis: separation of aspects (or subtopics) of opinions, understanding of sentiments, and assessment of quality of opinions. These dimensions form three key components in an integrated opinion summarization system. Accordingly, this thesis makes contributions in proposing novel and general computational techniques for three synergistic tasks: (1) integrating relevant opinions from all kinds of Web 2.0 sources and organizing them along different aspects of the topic which not only serves as a semantic grouping of opinions but also facilitates user navigation into the huge opinion space; (2) inferring the sentiments in the opinions with respect to different aspects and different opinion holders, so as to provide the users with a more detailed and informed multi-perspective view of the opinions; and (3) improving the prediction of opinion quality which critically decides the usefulness of the information extracted from the opinions.

We focus on general and robust methods which require minimal human supervision so as to make the automated methods applicable to a wide range of topics and scalable to large

amounts of opinions. This focus differentiates this thesis from work that is fine-tuned or well-trained for particular domains but are not easily adaptable to new domains. Our main idea is to exploit many naturally available resources, such as structured ontologies and social networks, which serve as indirect signals and guidance for generating opinion summaries. Along this line, our proposed techniques have been shown to be effective and general enough to be applied for potentially many interesting applications in multiple domains, such as business intelligence and political science.

To my parents.

Acknowledgments

First and foremost, I would like to express my deepest appreciation to my advisor Professor ChengXiang Zhai. From the very begining, it is Cheng who inspired my interest in text information management with his great passion and broad vision in the area. I have learned from him not only to conduct high quality research but also to always maintain a rigorous attitude toward research. Cheng's dedication to research problems that will make real impact has deeply influenced me when choosing my thesis topic. This thesis would not have been possible without his constant guidance and encouragement, and working with him has made years of doctoral study much more enjoyable.

My great gratitude goes to all the other thesis committee members, Professor Jiawei Han, Professor Dan Roth, Professor Panayoitis Tsaparas for their generous time and commitment. Their constructive comments and suggestions made this thesis more complete and accurate. Special thanks to Professor Panayoitis Tsaparas from University of Ioannina, who made a special trip to UIUC for my final defense.

I have been blessed to receive much help from many collaborators, colleagues, and friends at UIUC. I would like to express my thanks to the members of the TIMan Group for many valuable discussions and help, especially, Hui Fang, Tao Tao, Xuehua Shen, Qiaozhu Mei, Xuanhui Wang, Jing Jiang, Bin Tan, Xu Ling, Xin He, Maryam Karimzadehgan, Alexander Kotov, V.G. Vinod Vydiswaran, Yuanhua Lv, Duo Zhang, Hyun Duk Kim, Hongning Wang, Huizhong Duan, Kavita Ganesan, and Dae Hoon Park. I would also like to thank many other members of the DAIS Group, especially Dong Xin, Hong Cheng, Deng Cai, Tao Cheng, Tianyi Wu, Jing Gao, Bolin Ding, Zhenhui Li, Yizhou Sun, Rui Li, Yunliang Jiang and

Bo Zhao. We had many interesting discussions as well as sharing many happy memories together. My UIUC llife was much healthier with much joyful time spent with my "gym buddies": Ying Huang, Zheng Zeng, Yan Gao and Yong Yang. Many thanks to them.

Lastly, and most importantly, I am grateful to my parents, whose love and support has encouraged me to overcome the difficulties to complete my doctoral study. To them I dedicate this thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

"What do people think?" (or public opinion) has always been an important and often indispensable piece of information during all kinds of decision-making processes. For example, in the business domain, users need opinions from other customers in order to make their choice of products or services [15, 33] while the companies also want such opinions for their marketing decisions [8]. Another example is in the political domain, where voters are seeking (e.g., [72]) and influenced [30] by others' opinions about political campaigns, elections, and government. At the same time, there is a also known linkage between public opinion and the action of political policy and decision makers [41]. While in the past we need to set up surveys and polls to collect people's opinion, nowadays, with the increasing popularity of Web 2.0 applications, more and more people actively express their opinions on the Web in various ways: customers review millions of products and services on Amazon[1], Yelp[2], and TripAdvisor[3]; patients and families discuss their experiences about various diseases and drugs in medical forums, e.g., HealthBoards[4]; voters comment on political figures, policies and campaigns in their personal blogs. Such wide coverage of topics and abundance of users make the Web an extremely valuable source for mining people's opinions about all kinds of topics.

However, it is very difficult for the users to make sense and extract useful information out of a large number of online opinions for their tasks or decisions. This is because that when

[1] www.amazon.com
[2] www.yelp.com
[3] www.tripadvisor.com
[4] www.healthboards.com

searching for people's opinions about a specific topic of interest (e.g. iPhone), users expect comprehensive results (which is implied by "*public* opinion") instead of only a few most relevant results. Thus, using search engines like Google can only accomplish the first step of gathering or collecting opinions relevant to a give topic. However, since search engines present the opinion results in the form of a ranked list, it is still almost impossible for the user to laboriously go through the long list. This thesis focuses on designing automatic methods to help users in further steps of *interpreting* and *validating* the collected opinions so that users can apply the information for their tasks and decisions.

The difficulty of interpreting online opinions lies in the fact that they are usually expressed as unstructured text containing complicated semantics. Using the iPhone example, we can see people comment on different *aspects* of iPhone (e.g., screen quality or phone reception) and express different *sentiment* toward the aspects (e.g. positive as in "screen is absolutely crystal clear" or negative as in "reception is unbearable"). Additionally, the *quality* or trustworthiness of online opinions varies a lot. Some opinions are comprehensive and trust-able while others are not helpful at all or even spam.

To help users interpret and digest huge amounts of opinions for any given topic, in this thesis, we study a new problem of opinion integration and summarization. A high level overview picture for this thesis is given in Figure 1.1. We propose an envisioned integrated summary for a given topic, and there are three important dimensions automatically extracted from opinions: the aspects, the sentiment, and the quality. To achieve this kind of output, we need to first identify different aspects in the diverse and dynamic opinions which help users easily navigate the large opinion space. Second, it is most useful to present sentiment polarity or opposing views to users, because such kinds of semantics underlining the opinions are inherently interesting, important, and differentiate opinions from traditional fact data. Third, the usefulness of information in opinions depends highly on its quality, so measuring and controlling the opinion quality provides a fundamental basis for other types of analysis. Such a summary would enable users a number of semantically meaningful operations. For

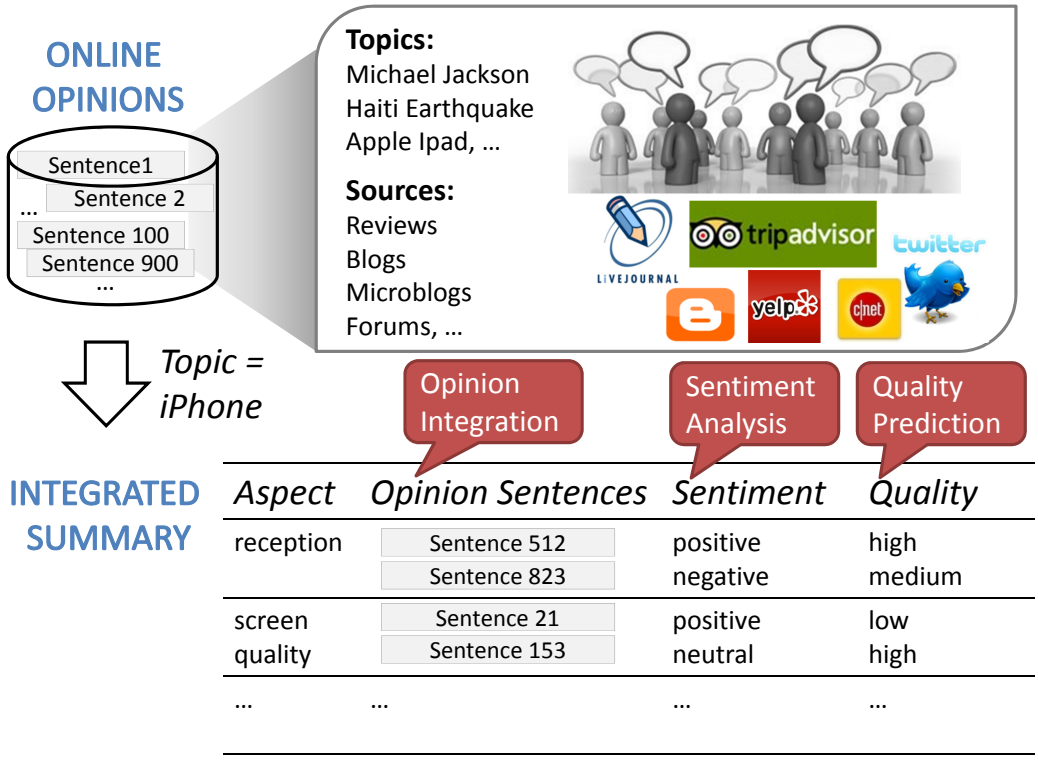


Figure 1.1: Thesis Overview

instance, the user can choose a subset of aspects she is mostly interested in so as to narrow down the information space. e.g., a businessman may want to read only opinions about the reception of iPhone. The user can also filter or rank aspects based on people's sentiment, e.g., display the most praised/positive aspect and the most criticized/negative one. Last but not least, the user is able to filter based on the quality of opinions, e.g., exclude opinions whose quality is in the lowest 10%. With the help of such a summary instead of a simple ranked list of results, users can identify the most useful information from the opinions more quickly and easily.

A major novelty of this thesis lies in the emphasis on developing *general* and *robust* techniques to generate such integrated summary effectively for *arbitrary* topics, such as political figures, events, products, services, companies, or brands. A significant advantage of general techniques over specialized techniques for particular domains or opinion summarization

problems is that a general method can be easily applied to many interesting applications in different domains, thus having broad impact. There is existing work that has tried to generate similar output for opinions (e.g., [34, 70]), but most of the techniques proposed are designed specifically for product reviews, and thus are restricted. In contrast, we focus on robust and effective methods which require minimal human-labeled training data and minimal manually encoded domain knowledge, so that the methods are applicable to a wide range of topics. This focus differentiates this thesis from most work in the NLP community that is usually well-trained or fine-tuned for particular domains, but not easily adaptable to new domains where new training data need to be obtained, or new heuristics need to be designed. Our main idea is to leverage existing resources that are naturally available on the Web in large amounts and in broad domains. These resources inherently carry topic specific information, thus can reduce the dependence on human supervision and domain specific heuristics. In this thesis, we will propose new methods that can effectively utilize these useful but indirect signals from different kinds of resources for generating integrated summary. Along this direction, this thesis includes the following three synergistic directions.

Opinion Integration: To understand people's opinions for any given topic, the first step is to integrate relevant opinions from all kinds of sources and organize them in a meaningful way. User generated opinions pose unique challenges, because opinions in blogs and forums are usually fragmented, scattered around, and buried among other off-topic content as well as being diverse and dynamic. To solve these challenges, we propose to leverage well-structured resources available on the Internet and organize scattered opinions along different aspects of the topic which not only serves as a semantic grouping of opinions but also facilitates user navigation into the huge opinion space. As the first work exploring this idea [51], we used a well-structured overview article (as provided in many web sites, e.g. CNET and Wikipedia) as a "template" to guide the organization of the scattered opinions and proposed a general probabilistic method for the integration. Another resource we exploited is the available structured ontology/database [84], such as product specifications in Google's product search

and the open domain knowledge base in Freebase. The related entities/relations in the database are used as candidate aspect labels and then the most interesting ones are selected to capture the major representative scattered opinions.

Aspect Level Sentiment Analysis: After integrating opinions from different sources and organizing them into meaningful aspects, we looked into the problem of inferring the sentiments of the opinions with respect to each aspect, so as to provide the users with a more detailed and informed multi-perspective view of the opinions. Unlike most existing work on sentiment analysis that rely on training a classifier from human labeled data, we aim at minimizing human supervision and instead maximizing the utilization of easy-to-collect information. We first defined and studied the novel problem of decomposing the available overall sentiment ratings (such as the star ratings provided in Amazon and TripAdvisor reviews) into ratings on the different aspects [52]. We proposed two general approaches for this new problem, both un-supervised: Local Prediction uses the local information of the overall rating of one opinionated text to rate the phrases in that text; Global Prediction rates phrases based on aspect level rating classifiers which are learned from overall ratings of all opinion text. Evaluated on a data set of seller feedback comments from eBay, Global Prediction is shown to generate more accurate rating prediction, but Local Prediction is sufficient at predicting a few representative phrases in each aspect. Later, we also included other different sources of information (in addition to overall sentiment ratings) for inferring aspect level sentiment, namely opinion text, sentiment lexicon and synonym/antonym dictionaries. We proposed a novel optimization framework that effectively combine all the signals from multiple sources in a unified and principled way.

User Level Sentiment Analysis: So far we have analyzed sentiment by treating each piece of opinion text as equivalent while apparently the opinion holder is also very important. For example, in real life, whether a Computer Science PhD student is pro-choice in the abortion issue cannot be counted the same as whether President Obama is. It is the same in online communities. As time goes by, some regular users may develop a sense of

"virtual community" which can be considered as a type of hidden social network. However, such sense of virtual community can only be formed by accumulated effort of reading and participating in lots of online discussions. This is very difficult to achieve by ordinary readers and even occasional forum users, who can only capture a local view of the posts without any context or background information. To this end, we study user level sentiment analysis and propose a new problem of automatically discovering *opposing opinion networks* from online forum discussions, which is a latent social network with links based on user opinions on different topics. In particular, we want to automatically identify two sets of opposing users for each topic: a supporting group and an against group. This raises interesting challenges, which we addressed by combining textual content information (e.g. post content) and social network information (e.g. who says what and who talks to whom).

Opinion Quality Prediction: The rapid growth of opinion data in Web 2.0 applications comes at the price of wide variation in the quality which may compromise the usefulness of the information. Thus, assessing opinion quality is a pressing challenge for opinion integration and analysis. To this end, we studied the problem of predicting the quality of user generated reviews [86]. Existing solutions employ supervised learning techniques and treat each review as a stand-alone text document. However, in order to learn a good prediction function, such supervised methods require a lot of labeled data, which is expensive to obtain. To solve this problem, we exploited social context, particularly reviewers identities and social network in a novel generic framework which adds graph-based regularization constraints to the text-based predictor. This approach can effectively use the social context information available for large amounts of *unlabeled* reviews. Experiments within a real commerce portal demonstrate that using social contextual information can effectively improve the accuracy of review quality prediction especially when the available training data is limited.

To the best of our knowledge, this thesis is the first systematic study on opinion integration and summarization. In particular, we aim at effective and robust methods that utilize

naturally available resources without requiring a lot of human labeled data, and thus are potentially applicable to different topics or domains. In this way, our work can easily lead to many interesting applications, ranging from business intelligence to political science.

Chapter 2

Related Work

This thesis is closely related to opinion mining, summarization and analysis which has attracted much attention recently. In this chapter, we review related work in existing literature.

2.1 Opinion Integration

Traditional text summarization techniques typically generate an *unstructured* list of sentences as summary, which is only effective for topics with very specific definition, such as a news story covered in several newspapers. Since online opinions are very diverse and dynamic, recent work has shown the usefulness of generating a *structured summary* of all the opinions about a topic which reveals representative opinions on *different aspects* of the topic and facilitates navigation into the huge opinion space. To this end, aspect summarization, i.e., structured opinion summarization over topical aspects, has attracted much attention recently [47, 70, 56, 77, 78].

A major challenge in producing such a structured organization or summary is how to generate these aspects for an *arbitrary* topic (e.g., products, political figures, policies, etc.). Text clustering is a traditional way of generating aspects of a text collection. In [45], the authors evaluated different clustering methods used in search result clustering and demonstrated that it is useful for interactive search. Zeng et. al [87] used supervised learning method to extract salient phrases among the search results, and group them into clusters according to the extracted phrases. Some work [32, 58, 79] used generative models to discover the latent aspects of the given texts. Some other work [47, 70] used frequent-pattern

or association rule mining method to find related aspects to a given product. After that, meaningful and prominent phrases need to be selected to represent the aspects, e.g. [92, 59]. However, these methods suffer from the problem of producing trivial aspects. Consequently, some of the aspects generated are very difficult to interpret [13].

In this thesis, we propose a novel kind of approach for organizing scattered opinions into meaningful aspects, that is, to leverage well-structured resources available on the Internet, including well-structured overview articles (in Section 3.1) and structured ontologies (in Section 3.2). Ontology is used in [11] but only for mapping product features. One close work [73] also uses well-written articles for structured summarization, but it requires a relatively large amount of training data in the given domain. In comparison, our work only needs one overview article or the ontology information for the given topic, which is much easier to obtain from resources such as Wikipedia and Freebase.

2.2 Aspect Level Sentiment Analysis

Analysis of the *overall sentiment* of review text has been extensively studied. (See [67] for a detailed review) Related research started from a definition of binary classification of a given piece of text into the positive or negative class [68, 18, 20, 81, 69, 16, 42]. Later, the definition is generalized into a multi-point rating scale [66, 27]. Many approaches have been proposed to solve the problem, including supervised, un-supervised, and semi-supervised approaches, but they all attempt to predict an *overall* sentiment class or rating to a review, which is not our focus.

Another line of work is to create a sentiment lexicon (i.e., words or phrases with sentiment scores assigned) in an unsupervised manner [29, 82, 40, 31, 64, 61, 37, 28]. Such sentiment lexicon can be used in many sentiment-related applications. There is no general-purpose sentiment lexicon that can work well for every domain or topic, because sentiments of words are well known to be domain dependent [82]. Indeed, domain adapted sentiment lexicons

have been shown to improve task performance in a number of applications, including opinion retrieval [63, 37], and expression level sentiment classification [14]. In those automatic methods, it is usually assumed that seed words with known polarity or a general-purpose sentiment lexicon is provided, whose polarity will be propagated to the unknown sentiment polarity of other words. Different heuristics as the propagation strategy have been proposed in existing work. Some are based on linguistic heuristics in the context [29, 40]. For example, two words linked by "but"-like conjunctions are most likely to be in opposite polarities, while conjunctions like "and" are evidences for words in the same polarity. Some works [64, 61] assume polarities of two words are correlated with their morphological relations and/or synonymy relations in thesaurus. Another popular type of methods, suggested by Turney [82], is to decide the polarity of a word or phrase by comparing whether it has a greater tendency to co-occur with the word "poor" (in a context window) or with the word "excellent" as measured by point-wise mutual information. Yet another kind of approaches exploit the association between words and expression-level or document-level sentiment [14, 83]. However, few of them consider the problem that even the same word in the same domain may indicate different polarities with respect to different aspects.

Since an online opinions usually contains multiple sentiment polarities on multiple aspects, some recent work has started to predict the *aspect level ratings* instead of one overall rating. A recent human evaluation [46] indicates that sentiment informed summaries are strongly preferred over non-sentiment baselines which shows the usefulness of modeling sentiment and aspects when summarizing opinions. Snyder et al. [74] show that modeling the dependencies among aspects using good grief algorithm can improve the prediction of aspect ratings. In [77], Titov et al. propose to extract aspects and predict the corresponding ratings simultaneously: they employ topics to describe aspects and incorporate a regression model fed by the ground-truth ratings. However, they are all in the supervised framework, i.e. assuming the aspect ratings are provided in the data. In comparison, we assume aspect level sentiments are latent, which is a more general and more realistic scenario.

Review mining is another line of relevant research that involves fine-grained sentiment analysis. Hu and Liu [35] apply association mining to extract product features and decide the polarity the opinions using a seed set of adjective expanded via WordNet, but there is no attempt to cluster the aspects, so "battery", "batteries", "power" would result in separate aspects/features. A similar work of OPINE [70], which outperforms Hu and Liu's system both in feature extraction and opinion polarity identification, shares the same problem. However, clustering (i.e., mapping different ways of mentioning the same concept to the same cluster) can be very important in domains where aspects are described using different vocabulary or misspellings are common as in online opinions and it is especially important for accurate aggregation of ratings.

In this thesis, we propose to infer aspect level sentiments using all the naturally available resources. In Section 4.1, we start with a new problem as inferring aspect level sentiment ratings from only the opinion text and associated overall sentiment ratings. Then, in Section 4.2, we further improve it by including more resources, i.e., general-purpose sentiment lexicon, thesaurus of synonyms/antonyms, and linguistic heuristics. There are some recent work that combines more than one resource (i.e. linguistic heuristics and synonym/antonym rules) [21], but still in an ad-hoc rule-based manner which solves possible conflicting polarities by simple majority voting. To the best of our knowledge, there is no existing method that can effectively combine all kinds of resources for inferring aspect level sentiments in a unified framework, as proposed in this thesis.

2.3 User Level Sentiment Analysis

There are fewer sentiment analysis work on the user level analysis. The natural resource to exploit here is the social relations among the users in addition to the opinion text they produced. For example, Galley et al. [23] described a supervised Maximum Entropy classification of utterances into Agreement/Disagreement using lexical, durational, structural

features, sequence information. Agrawal et al. [7] proposed a method to classify the supporting/opposing position of users based on their observation that reply activities usually show disagreement with previous authors. However, these previous work either use the link information without using any of the content information or use the context content without much consideration of users social interaction. Instead, our work has shown that the combination of content analysis and social network analysis is most powerful.

Previous work that has also shown this power include [76], which introduced consistency constraints that a single speaker stays the same position for the classifying participants positions in debates. Their study is in the supervised setting, adding constraints to SVM classifier, while we are interested in the more difficult unsupervised setting. The work closest to ours is [62] which proposed an unsupervised approach: first use a rule-based method to classify the replies into agree, disagree, and neutral, then use max cut to classify the users into supporting or opposing parties. However, the use of the rule-based classifier limit the performance of their method. This is because their predefined pattern dictionary can hardly cover all cases when we apply to very different types of forums/issues which involve different vocabulary and slangs. Instead, our methods fully exploit the forum data itself using both textual analysis and social network analysis, so that they can automatically adapt to different types of topics or forums.

2.4 Opinion Quality

The problem of assessing the quality of user-generated content is recently attracting increasing attention. Most previous work [91, 43, 48, 25, 49, 80] has typically focused on automatically determining the quality (or helpfulness/utility/trustworthyness) of reviews by using textual features. The problem of determining review quality is formulated as a classification or regression problem with users' votes serving as the ground-truth. In this context, Zhang and Varadarajan [91] found that shallow syntactic features from the text of

reviews are most useful, while review length seems weakly correlated with review quality. In addition to textual features, Kim et al. [43] included metadata features including ratings given to an item under review and concluded that review length and the number of stars in product rating are most helpful within their SVM regression model. Ghose and Ipeirotis [25] combined econometric models with textual subjectivity analysis and demonstrated evidence that extreme reviews are considered to be most helpful. In [49], the authors incorporated reviewers' expertise and review timeliness in addition to the writing style of the review in a non-linear regression model. In our work, we extend previous work by adding a novel resource in order to assess review quality, i.e., the author and social network information .

Although user votes can be helpful as ground-truth data, Liu et al [48] identified a discrepancy between votes coming from `Amazon.com` and votes coming from an independent study. More specifically, they identified a "rich-get-richer" effect, where reviews accumulate votes more quickly depending on the number of votes they already have. This observation further enhances our motivation to automatically determine the quality of reviews in order to avoid such biases. Danescu-Niculescu-Mizil et al. [17] showed that the perceived helpfulness of a review depends not only on its content but also on the other reviews of the same product. A recent paper [80] took an un-supervised approach to finding the most helpful book reviews. Although their method is shown to outperform users' votes, it is evaluated on only 12 books and thus is not clear whether it is robust and generalizable.

The problem of assessing the quality of user-generated data is also critical in domains other than reviews. For example, previous works [6, 10] focused on assessing the quality of postings within the community question/answering domain. The work in [6] combines textual features with user and community meta-data features for assessing the quality of questions and answers. In [10], the authors propose a co-training idea that jointly models the quality of the author and the review. However, their work does not explicitly model user relationships, bur rather uses all community information for exacting features.

We use graph regularization to incorporate social context to review quality prediction.

Regularization using graphs has appeared as a type of effective method in the semi-supervised learning literature [94]. The interested reader may examine [93, 95, 9]. The resulting formulation is usually a well-formed convex optimization problem which has a unique and efficiently computable solution. These types of graph regularization methods have been successfully applied in Web-page categorization [90] and Web spam detection [5]. In both cases, the link structure among Web pages is nicely exploited by the regularization which, in most cases, has improved the predictive accuracy for the problem at hand. Recently, Mei et al. [54] propose to enhance topic models by regularizing on a contextual graph structure. In our scenario, the social network of the reviewers defines the context, and we exploit it to enhance review quality prediction.

Chapter 3

Opinion Integration

The explosive growth of online opinions raises interesting challenges for opinion integration and summarization. It is especially interesting to integrate and summarize scattered opinions in blog articles and forums as they tend to represent the general opinions of a large number of people and get refreshed quickly as people dynamically generate new content, making them valuable for understanding the current views of a topic. However, opinions in blogs and forums are usually fragmental, scattered around, and buried among other off-topic content, so it is quite challenging to organize them in a meaningful way. Recent work [56, 77, 78] has shown the usefulness of generating a structured summary of opinions, in which related opinions are grouped into topical aspects with explicit labeling of all the aspects. A major challenge in producing such a structured organization or summary is how to generate these aspects for an *arbitrary* topic. In this chapter, we propose to leverage existing structured resources: (1) overview articles, as provided in many web sites, e.g. CNET and Wikipedia, and (2) structured ontology, such as product specifications in Google's product search and the open domain knowledge base in Freebase. Both types of resources are freely available on the Internet and are constantly growing as people continuously contribute. Note that, the two kinds of resources have only been explored independently so far, but it is possible to combine the resources for generating more effective aspects, which is an interesting topic for future work.

3.1 Exploiting Overview Articles

3.1.1 Overview

In general, for any given topic (e.g., a product), there are often two kinds of opinions: opinions expressed in some well-structured and comprehensive review typically written by some expert about the topic, and fragmental opinions scattered around in all kinds of sources such as blog articles and forums. For convenience of discussion, we will refer to the first as *expert opinions* and the second as *ordinary opinions*. The expert opinions are relatively easy for a user to access through some opinion search web site such as CNET. Because a comprehensive product review is often written carefully, it is also easy for a user to digest expert opinions. However, finding, integrating, and digesting ordinary opinions poses significant challenges as they are scattered in many different sources, and are generally fragmental and not well structured. While expert opinions are clearly very useful, they may be biased and often out of date after a while. In contrast, ordinary opinions tend to cover the opinions of a large number of people and get refreshed quickly as people dynamically generate new content. For example, a query "iPhone" returns 330,431 matches in Google's blogsearch (as of Nov. 1, 2007), suggesting that there are many opinions expressed about iPhone in blog articles within a short period of time since it hit the market. To enable a user to benefit from both kinds of opinions, it is thus necessary to automatically combine these two kinds of opinions together and present an integrated opinion summary to a user.

To the best of our knowledge, such an integration problem has not been studied in the existing work. In this section, we study how to integrate a well-written expert review about an arbitrary topic with many ordinary opinions expressed in a text collection such as blog articles. We propose a general method to solve this integration problem in three steps: (1) extract ordinary opinions from text using information retrieval; (2) summarize and align the extracted opinions to the expert review to integrate the opinions; (3) further distinguish ordinary opinions that are similar to expert opinions from those that are not. Our main idea

is to take advantage of the high readability of the expert review to structure the unorganized ordinary opinions while at the same time summarizing the ordinary opinions to extract *representative* opinions using the expert review as guidance. From the viewpoint of text data mining, we are essentially to use the expert review as a "template" to mine text data for ordinary opinions. The first step in our approach can be implemented with a direct application of information retrieval techniques. Implementing the second and third steps involves special challenges. In particular, without any training data, it is unclear how we should align ordinary opinions to an expert review and separate similar and supplementary opinions. We propose a semi-supervised topic modeling approach to solve these challenges. Specifically, we cast the expert review as a prior in a probabilistic topic model (i.e., PLSA[32]) and fit the model to the text collection with the ordinary opinions with Maximum A Posteriori (MAP) estimation. With the estimated probabilistic model, we can then naturally obtain alignments of opinions as well as additional ordinary opinions that cannot be well-aligned with the expert review. We use a similar model to separate similar opinions and supplementary opinions.

We evaluate our method on integrating opinions about two quite different topics. One is a popular product "iPhone", and the other is a popular political figure Barack Obama. Experimental results show that our method can effectively integrate the expert review (a produce review from CNET for iPhone and a short biography from Wikipedia for Barack Obama) with ordinary opinions from blog articles.

3.1.2 Problem Definition

In this section, we define the novel problem of opinion integration.

Given an expert review about a topic T (e.g., "iPhone" or "Barack Obama") and a collection of text articles (e.g., blog articles), our goal is to extract opinions from text articles and integrate them with those in the expert review to form an integrated opinion summary.

The expert review is generally well-written and coherent, thus we can view it as a se-

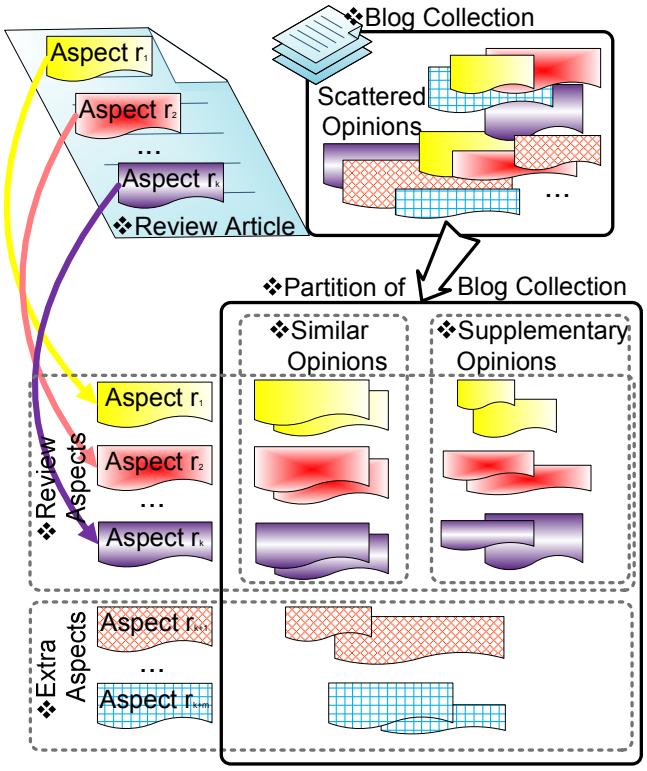


Figure 3.1: Problem Setup

quence of semantically coherent segments, where a segment could be a sentence, a paragraph, or other meaningful text segments (e.g., paragraphs corresponding to product features) available in some semi-structured review. Formally, we denote the expert review by $R = \{r_1, ..., r_k\}$ where r_i is a segment. Since we can always treat a sentence as a segment, this definition is quite general.

The text collection is a set of text documents where ordinary opinions are expressed and can be represented as $\mathcal{C} = \{d_1, ..., d_{|C|}\}$ where $d_i = (s_{i1}, ..., s_{i|d_i|})$ is a document and s_{ij} is a sentence. To support opinion integration in a general and robust manner, we do not rely on extra knowledge to segment documents to obtain opinion regions; instead, we treat each sentence as an opinion unit. Since a sentence has a well-defined meaning, this assumption is reasonable. To help a user interpret any opinion sentence, in real applications we would link each extracted opinion sentence back to the original document to facilitate navigating into the original document and obtaining context of an opinion.

We would like our integrated opinion summary to include both opinions in the expert review and the most representative opinions in the text collection. Since an expert review is in general well written, we keep their original form and leverage its structure to organize the ordinary opinions extracted from text. To quantify the representativeness of an ordinary opinion sentence, we will compute a "support value" for each extracted ordinary opinion sentence. Specifically, we would like to partition the extracted ordinary opinion sentences into groups that can be potentially aligned with all the review segments $r_1, ..., r_k$. Naturally, there may also be some groups with ordinary opinion sentences that are not alignable with any expert opinion segment; but these opinions can be very useful for augmenting the expert review with additional opinions.

Furthermore, for opinion sentences aligned to a review segment r_i, we would like to further separate those that are similar to r_i from those that are supplementary for r_i; such separation can allow a user to digest the integrated opinions more easily.

Finally, if r_i has multiple sentences, we can further align each ordinary opinion sentence (both "similar" and "supplementary") with a sentence in r_i to increase the readability.

This problem setup is illustrated in Figure 3.1. We now define the problem more formally.

Definition (Representative Opinion (RO)) A *representative opinion(RO)* is an ordinary opinion sentence extracted from the text collection with a support value. Formally, we denote it by $o_{ij} = (\beta, s_{ij})$ where $\beta \in [1, +\infty)$ is a support value indicating how many sentences this opinion sentence can represent, and s_{ij} is a sentence in document d_i.

Since ordinary opinions tend to contain redundant content and we are primarily interested in extracting representative opinions, the support can be very useful to assess the representativeness of an extracted opinion sentence.

Let $RO(\mathcal{C})$ be all the possible representative opinion sentences in $\mathcal{C}$. We can now define the integrated opinion summary that we would like to generate as follows.

Definition (Integrated Opinion Summary) An *integrated opinion summary* of R and

$\mathcal{C}$ is a tuple $(R, S^{sim}, S^{supp}, S^{extra})$ where (1) R is the given expert review; (2) $S^{sim} = \{S_1^{sim}, ..., S_k^{sim}\}$ and $S^{supp} = \{S_1^{supp}, ..., S_k^{supp}\}$ are similar and supplementary representative opinion sentences, respectively, that can be aligned to R, and $S_i^{sim}, S_j^{supp} \subset RO(\mathcal{C})$ are sets of representative opinion sentences; (3) $S^{extra} \subset RO(\mathcal{C})$ is a set of extra representative opinion sentences that cannot be aligned with R.

Note that we define "opinion" broadly as covering all the discussion about a topic in opinionate sources such as blog spaces and forums. The notion of "opinion" is quite vague; we adopt this broad definition to ensure generality of the problem set up and its solutions. In addition, any existing sentiment analysis technique could be applied as a post-processing step. But since we only focus on the integration problem in this paper, we will not cover sentiment analysis.

3.1.3 Overview of Proposed Approach

The opinion integration problem as defined in the previous section is quite different from any existing problem setup for opinion extraction and summarization, and it presents some unique challenges: (1) How can we extract representative opinion sentences with support information? (2) How can we distinguish alignable opinions from non-alignable opinions? (3) For any given expert review segment, how can we distinguish similar opinions from those that are supplementary? (4) In the case when a review segment r_i has multiple sentences, how can we align a representative opinion to a sentence in r_i? In this section, we present our overall approach to solving all these challenges, leaving a detailed presentation to the next section.

At a high level, our approach primarily consists of two stages and an optional third stage: In the first stage, we retrieve only the relevant opinion sentences from $\mathcal{C}$ using the topic description T as a query. Let $\mathcal{C}_O$ be the set of all the retrieved relevant opinion sentences. In the second stage, we use probabilistic topic models to cluster sentences in $\mathcal{C}_O$

and obtain S^{sim}, S^{supp} and S^{extra}. When r_i has multiple sentences, we have a third stage, in which we again use information retrieval techniques to align any extracted representative opinion to a sentence of r_i. We now describe each of the three stages in detail.

The purpose of the first stage is to filter out irrelevant sentences and opinions in our collection. This can be done by using the topic description as a keyword query to retrieve opinion sentences relevant to the topic of interest. In general, we may use any retrieval method. In this paper, we used a standard language modeling approach (i.e., the KL-divergence retrieval model [88]). To ensure coverage of opinions, we perform pseudo feedback using some top-ranked sentences; the idea is to expand the original topic description query with additional words related to the topic so that we can further retrieve opinion sentences that do not necessarily match the original topic description T. After this retrieval stage, we obtain a set of relevant opinion sentences $\mathcal{C}_O$.

In the second stage, our main idea is to exploit a probabilistic topic model, i.e., Probabilistic Latent Semantic Analysis (PLSA) with conjugate prior [32, 57] to cluster opinion sentences in a special way so that there is precisely one cluster corresponding to each segment r_i in the expert review. These clusters collect opinion sentences that can be aligned with a review segment. There will also be some clusters that are not aligned with any review segments, and they are designed to collect extra opinions. Thus the model provides an elegant way to simultaneously partition opinions and align them to the expert review. Interestingly, the same model can also be adapted to further partition opinions aligned to a review segment into similar and supplementary opinions. Finally, a simplified version of the model (i.e., no prior, basic PLSA) can be used to cluster any group of sentences to extract representative opinion sentences. The support of a representative opinion is defined as the size of the cluster represented by the opinion sentences.

Note that what we need in this second stage is semi-supervised clustering in the sense that we would like to constrain many of the clusters so that they would correspond to the segments r_is in the expert review. Thus a direct application of any regular clustering algorithm would

not be able to solve our problem. Instead of doing clustering, we can also imagine using each expert review segment r_i as a query to retrieve similar sentences. However, it would be unclear how to choose a good cutoff point on the ranked list of retrieved results. Compared with these alternative approaches, PLSA with conjugate prior provides a more principled and unified way to tackle all the challenges.

In the optional third stage, we have a review segment r_i with multiple sentences and we would like to align all extracted representative opinions to the sentences in r_i. This can be achieved by using each representative opinion as a query and retrieve sentences in r_i. Once again, in general, any retrieval method can be used. In this paper, we again used the KL-divergence retrieval method.

From the discussion above, it is clear that we leverage both information retrieval techniques and text mining techniques (i.e., PLSA), and our main technical contributions lie in the second stage where we repeatedly exploit semi-supervised topic modeling to extract and integrate opinions. We describe this step in more detail in the next section.

3.1.4 Semi-Supervised PLSA for Opinion Integration

Probabilistic latent semantic analysis (PLSA) [32] and its extensions [89, 60, 57] have recently been applied to many text mining problems with promising results. Our work adds to this line yet another novel use of such models for opinion integration.

As in most topic models, our general idea is to use a unigram language model (i.e., a multinomial word distribution) to model a topic. For example, a distribution that assigns high probabilities to words such as "iPhone", "battery", "life", "hour", would suggest a topic such as "battery life of iPhone." In order to identify multiple topics in text, we would fit a mixture model involving multiple multinomial distributions to our text data and try to figure out how to set the parameters of the multiple word distributions so that we can maximize the likelihood of the text data. Intuitively, if two words tend to co-occur with each other and one word is assigned a high probability, then the other word generally should also

be assigned a high probability to maximize the data likelihood. Thus this kind of model generally captures the co-occurrences of words and can help cluster the words based on co-occurrences.

In order to apply this kind of model to our opinion integration problem, we assume that each expert review segment corresponds to a unigram language model which would capture all ordinary opinion sentences that can be aligned with a review segment. Furthermore, we introduce a certain number of unigram language models to capture the extra opinions. We then fit the mixture model to $\mathcal{C}_O$, i.e., the set of all the relevant opinion sentences generated using information retrieval as described in the previous section. Once the parameters are estimated, they can be used to group sentences into different aspects corresponding to the different review segments and extra aspects corresponding to extra opinions. We now present our mixture model in detail.

Basic PLSA

We first present the basic PLSA model as described in [89]. Intuitively, the words in our text collection $\mathcal{C}_O$ can be classified into two categories (1) background words that are of relatively high frequency in the whole collection. For example, in the collection of topic "iPhone", words like "iPhone", "Apple" are considered as background words. (2) words related to different aspects which we are interested in. So we define $k+1$ unigram language models: θ_B as the background model to capture the background words, $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$ as k ($k = k^{expert} + k^{extra}$) theme models, each capturing one aspect of the topic and corresponding to the k^{expert} review segments $r_1, ..., r_k$ or k^{extra} extra aspects. A document d in $\mathcal{C}_O$ (in our problem it is actually a sentence) can then be regarded as a sample of the following mixture model.

$$p_d(w) = \lambda_B p(w|\theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} [\pi_{d,j} p(w|\theta_j)] \tag{3.1}$$

where w is a word, $\pi_{d,j}$ is a document-specific mixing weight for the j-th aspect ($\sum_{j=1}^{k} \pi_{d,j} = 1$), and λ_B is the mixing weight of the background model θ_B. The log-likelihood of the collection $\mathcal{C}_O$ is

$$\log p(\mathcal{C}_O|\Lambda) = \sum_{d\in\mathcal{C}_O} \sum_{w\in V} \{c(w,d)\times \log(\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}[\pi_{d,j} p(w|\theta_j)]\} \tag{3.2}$$

where V is the set of all the words (i.e., vocabulary), $c(w,d)$ is the count of word w in document d, and Λ is the set of all model parameters. The purpose of using a background model is to "force" clustering to be done based on more discriminative words, leading to more informative and more discriminative theme models.

The model can be estimated using any estimator. For example, the Expectation-Maximization (EM) algorithm [19] can be used to compute a maximum likelihood estimate with the following updating formulas: (z are the introduced hidden variables)

$$\begin{aligned}
p(z_{d,w,j}) &= \frac{(1-\lambda_B)\pi_{d,j}^{(n)} p^{(n)}(w|\theta_j)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j'=1}^{k} \pi_{d,j'}^{(n)} p^{(n)}(w|\theta'_j)} \\
p(z_{d,w,B}) &= \frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j'=1}^{k} \pi_{d,j'}^{(n)} p^{(n)}(w|\theta'_j)} \\
\pi_{d,j}^{(n+1)} &= \frac{\sum_{w\in V} c(w,d) p(z_{d,w,j})}{\sum_{j'}\sum_{w\in V} c(w,d) p(z_{d,w,j'})} \\
p^{(n+1)}(w|\theta_j) &= \frac{\sum_{d\in\mathcal{C}_O} c(w,d) p(z_{d,w,j})}{\sum_{w'\in V}\sum_{d\in\mathcal{C}_O} c(w,d) p(z_{d,w',j})}
\end{aligned}$$

Semi-supervised PLSA

We could have directly applied the basic PLSA to extract topics from $\mathcal{C}_O$. However, the extracted topics in this way would generally not be well-aligned to the expert review. In order to ensure alignment, we would like to "force" some of the multinomial distribution component models (i.e., language models) to be "aligned" with all the segments in the expert review. In probabilistic models, this can be achieved by extending the basic PLSA to incorporate

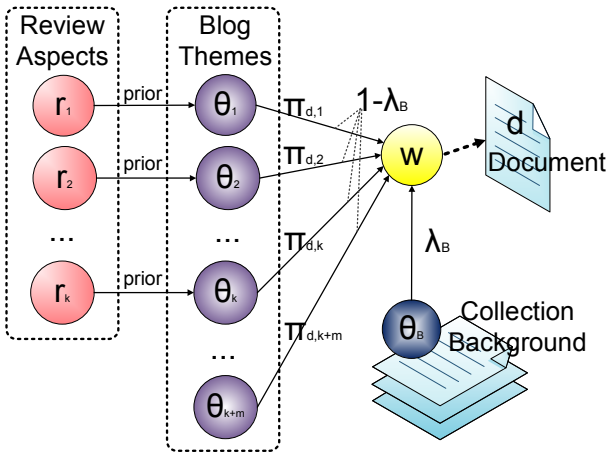


Figure 3.2: Generation Process of a Word

a conjugate prior defined based on the expert review segments and using the Maximum A Posteriori (MAP) esatimator instead of the Maximum Likelihood estimator as we did in the basic PLSA. Intuitively, a prior defined based on an expert review segment would tend to make the corresponding language model similar to the empirical word distribution in the review segment, thus the language model would tend to attract opinion sentences in $\mathcal{C}_O$ that are similar to the expert review segment. This ensures the alignment of the extracted opinions with the original review segment.

Specifically, we build a unigram language model $\{p(w|r_j)\}_{w \in V}$ for each expert review segment r_j ($j \in \{1, ..., k\}$) and define a Dirichlet prior (i.e., a conjugate prior for multinomial) on each multinomial distribution topic model, parameterized as $Dir(\{\sigma_j p(w|r_j)\} w \in V)$, where σ_j is a confidence parameter for the prior. Since we use a conjugate prior, σ_j can be interpreted as the "equivalent sample size" which means that the effect of adding the prior would be equivalent to adding $\sigma_j p(w|r_j)$pseudo counts for word w when we estimate the topic model $p(w|\theta_j)$. Figure 3.2 illustrates the generation process of a word W in such a semi-supervised PLSA where the prior serves as some "training data" to bias the clustering results.

The prior for all the parameters is given by

$$p(\Lambda) \propto \prod_{j=1}^{k+m} \prod_{w \in V} p(w|\theta_j)^{\sigma_j p(w|r_j)} \tag{3.3}$$

Generally we have $m > 0$, because we may want to find extra opinion topics other than the corresponding segments in the expert review. So we set $\sigma_j = 0$ for $k < j \leq k + m$.

With the prior defined above, we can then use the Maximum A Posteriori (MAP) estimator to estimate all the parameters as follows

$$\hat{\Lambda} = \arg\max_{\Lambda} p(\mathcal{C}_O|\Lambda)p(\Lambda) \tag{3.4}$$

The MAP estimate can be computed using essentially the same EM algorithm as presented above with only slightly different updating formula for the component language models. The new updating formula is:

$$p(w|\theta_j)^{(n+1)} = \frac{\sum_{d \in \mathcal{C}_O} c(w,d)p(z_{d,w,j}) + \sigma_j p(w|r_j)}{\sum_{w' \in V} \sum_{d' \in \mathcal{C}_O} c(w',d')p(z_{d',w',j}) + \sigma_j} \tag{3.5}$$

We can see that the main difference between this equation and the previous one for basic PLSA is that we now pool the counts of terms in the expert review segment with those from the opinion sentences in $\mathcal{C}_O$, which essentially allows the expert review to serve as "training data" for the corresponding opinion topic. This is why we call this model semi-supervised PLSA.

If we are highly confident of the aspects captured in the prior, we could empirically set a large σ_j. Otherwise, if we need to ensure the impact of the prior without being over restricted by the prior, some regularized estimation techniques are necessary. Following the similar idea of regularized estimation [75], we define a decay parameter η and a prior weight μ_j ($\mu_j \in [-1, 1]$ measures how much of an estimated topic is attributed to the given prior)

as

$$\mu_j = \frac{\sigma_j}{\sum_{w' \in V} \sum_{d' \in \mathcal{C}_O} c(w', d') p(z_{d',w',j}) + \sigma_j} \tag{3.6}$$

We start with a large σ_j (say 5000) (i.e., starting with perfectly alignable opinion models) and gradually decay it in each EM iteration using equation 3.7. We stop the decaying of σ_j when the weight of the prior μ_j is below some threshold δ (say 0.5). Decaying allows the model to gradually pick up words from $\mathcal{C}_O$, and the threadholding maintains the contribution of prior in the model. The new updating formulas are

$$\sigma_j^{(n+1)} = \begin{cases} \eta \sigma_j^{(n)} & \texttt{if } \mu_j > \delta \\ \\ \sigma_j^{(n)} & \texttt{if } \mu_j \leq \delta \end{cases} \tag{3.7}$$

$$p(w|\theta_j)^{(n+1)} = \frac{\sum_{d \in \mathcal{C}_O} c(w, d) p(z_{d,w,j}) + \sigma_j^{(n+1)} p(w|r_j)}{\sum_{w' \in V} \sum_{d' \in \mathcal{C}_O} c(w', d') p(z_{d',w',j}) + \sigma_j^{(n+1)}} \tag{3.8}$$

Overall Process

In this section, we describe how we use the semi-supervised topic model to achieve the three tasks in the second stage as defined in Section 3.1.3. We also summarize the computational complexity of the whole process.

Theme Extraction from Text Collection: We start from a topic T, a review $R = \{r_1, ..., r_k\}$ of k segments, a collection $\mathcal{C}_O = \{d_1, d_2, ..., d_N\}$ of opinion sentences closely relevant to T. We assume that $\mathcal{C}_O$ covers a number of themes each about one aspect of the topic T. We further assume that there are $k + m$ major themes in the collection, $\{\theta_1, \theta_2, ..., \theta_{k+m}\}$, each being characterized by a multinomial distribution over all the words in our vocabulary V (also known as a unigram language model or a topic model).

We propose to use review aspects as priors in the partition of $\mathcal{C}_O$ into aspects. We could have used the whole expert review segment to construct the priors. But if so, we could only get the opinions that are most similar to the review opinions. However, we would like

extract not only opinions supporting the review opinions but also supplementary opinions on the review aspect. So we use only the "aspect words" to estimate the prior. We use a simple heuristic: opinions are usually expressed in the form of adjectives, adverbs and verbs while aspect words are usually nouns. And we apply a Part-of-Speech tagger[1] on each review segment r_i and further filter out the opinion words to get a r'_i. The prior $\{p(w|r'_i)\}_{w\in V}$ is estimated by Maximum Likelihood:

$$p(w|r'_i) = \frac{c(w, r'_i)}{\sum_{w'\in V} c(w', r'_i)} \tag{3.9}$$

Given these priors constructed from the expert review $\{p(w|r'_i)\}_{w\in V}$, $i \in \{1, ..., k\}$, we could estimate the parameters for the semi-supervised topic model according to Section 3.1.4. After that, we have a set of theme models extracted from the text collection $\{\theta_i|i = 1, ..., k+m\}$, and we could group each sentence d_i in $\mathcal{C}_O$ into one of the $k + m$ themes by choosing the theme model with the largest probability of generating d_i:

$$g(d_i) = \arg\max_j p(d_i|\theta_j) = \arg\max_j \sum_{w\in V} c(w, d_i)p(w|\theta_j) \tag{3.10}$$

where $g(d_i) = j$ means that d_i is grouped into the jth theme model $\{p(w|\theta_j)\}_{w\in V}$. Now we have a partition of $\mathcal{C}_O$:

$$\mathcal{C}_O = \{S_i|i = 1, ..., k + m\} \tag{3.11}$$

where each S_i is a set of sentences $S_i = \{d_j|g(d_j) = i, d_j \in \mathcal{C}_O\}$ with the following two properties:

$$\mathcal{C}_O = \bigcup_{i=1}^{k+m} S_i \tag{3.12}$$

$$S_i \cap S_j = \emptyset \qquad \forall i, j \in \{1, ..., k + m\}\ , i \neq j \tag{3.13}$$

Thus each S_i, $i = 1, ..., k$, corresponds to the review aspect r_i and each S_j, $j = k +$

[1] http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=LBPPOS

$1, ..., k + m$, is the set of sentences that supplements the expert review with additional aspects. Parameter m, the number of additional aspects, is set empirically.

Further Separation of Opinions: In this subsection, we show how to further partition each S_i, $i = 1, ..., k$ into two parts:

$$S_i = \{S_i^{sim},\ S_i^{supp}\} \tag{3.14}$$

such that S_i^{sim} contains sentences that is similar to the opinions in the review while S_i^{supp} is a set of sentences that supplement the review opinions on the review aspect r_i.

We assume that each subset of sentences S_i, $i = 1, ..., k$, covers two themes captured by two subtopic models $\{p(w|\theta_i^{sim})\}_{w \in V}$ and $\{p(w|\theta_i^{supp})\}_{w \in V}$. We first construct a unigram language model $\{p(w|r_i)\}_{w \in V}$ from review segment r_i using both the feature words and opinion words. This model is used as a prior for extracting $\{p(w|\theta_i^{sim})\}_{w \in V}$. After that, we estimate the model parameters as described in Section 3.1.4. And then, we classify each sentence $d_j \in S_i$ into either S_i^{sim} or S_i^{supp} in the way similar to equation 3.10.

Generation of Summaries: So far, we have a meaningful partition over $\mathcal{C}_O$:

$$\mathcal{C}_O = \{S_1^{sim}, ..., S_k^{sim}\} \cup \{S_1^{supp}, ..., S_k^{supp}\} \cup \{S_{k+1}, ..., S_{k+m}\} \tag{3.15}$$

Now we need to further summarize each cluster P in the partition $P \in \{S_1^{sim}, ..., S_k^{sim}\} \cup \{S_1^{supp}, ..., S_k^{supp}\} \cup \{S_{k+1}, ..., S_{k+m}\}$ by extracting representative opinions $RO(P)$. We take a two-step approach.

In the first step, we try to remove the redundancy of sentences in P and group the similar opinion sentences together by unsupervised topic modeling. In detail, we use PLSA (without any prior) to do the clustering in P and set the number of clusters proportional to the size of P. After the clustering, we get a further partition of $P = \{P_1, ..., P_l\}$ where $l = |P|/c$ and c is a constant parameter that defines the average number of sentences in each cluster.

One representative sentence in P_i is selected by the similarity between the sentence and the cluster centroid (i.e. a word distribution) of P_i. If we define rs_i as the representative sentenced of P_i, and $\beta_i = |P_i|$ as the support, we have a representative opinion of P_i which is $o_i = (\beta_i, rs_i)$. Thus $RO(P) = \{o_1, o_2, ..., o_l\}$.

In the second step, we aim at providing some context information for each representative opinion o_i of P to help the user to better understand the opinion expressed. What we propose is to compare the similarity between opinion sentence rs_i and each review sentence in segment corresponding to P and assign rs_i to the review sentence with the highest similarity, which can be considered as the "centroid" of the cluster. For both steps, we use KL-Divergence as the similarity measure.

Computational Complexity: PLSA and semi-supervised PLSA have the same complexity: $O(I \cdot K(|V| + |W| + |\mathcal{C}|))$, where I is the number of EM iterations, K is the number of themes, $|V|$ is the vocabulary size, $|W|$ is the total number of words in the collection, $|\mathcal{C}|$ is the number of documents. Our whole process makes multiple invocations of PLSA/semi-supervised PLSA, and we suppose we use the same I across different invocations.

"Theme Extraction from Text Collection" makes one invocation of semi-supervised PLSA on the whole collection $\mathcal{C}_O$, where the number of cluster is $k + m$. So the complexity is $O(I \cdot (k + m) \cdot (|V| + |W| + |\mathcal{C}_O|) = O(I \cdot (k + m) \cdot |W|)$.

There are k invocations of semi-supervised PLSA in "Further Separation of Opinions", each on a subset of the collection $S_i (i = 1, ..., k)$ with only two clusters. And we know from equation 3.11 that $\bigcup_{i=1}^{k} S_i \subseteq \bigcup_{i=1}^{k+m} S_i = \mathcal{C}_O$. Suppose W_{S_i} is the total number of words in S_i. So the total complexity is $O(\sum_{S_i} I \cdot 2 \cdot (|V| + |W_{S_i}| + |S_i|))$ which in the worst case is $O(I \cdot 2 \cdot (k|V| + |W| + |\mathcal{C}_O|))$. Also, since the number of documents is usually much smaller than the total number of words, the complexity is essentially $O(I \cdot (k|V| + |W|))$.

Finally, "Generation of Summaries" makes $2k+m$ invocations of PLSA, each on a subset of the collection $P \in \{S_1^{sim}, ..., S_k^{sim}\} \cup \{S_1^{supp}, ..., S_k^{supp}\} \cup \{S_{k+1}, ..., S_{k+m}\} = \mathcal{C}_O$. In each invocation, the number of clusters is $\frac{|P|}{c}$, and W_P is the total number of words in P. So the

total complexity in this stage is $O(\sum_P I \cdot \frac{|P|}{c}(|V| + |W_P| + |P|))$, which in the worst case is $O(\frac{I}{c} \cdot (|\mathcal{C}_O| \cdot |V| + |\mathcal{C}_O| \cdot |W| + |\mathcal{C}_O|^2)) = O(\frac{I}{c} \cdot |\mathcal{C}_O| \cdot |W|)$.

Thus, our whole process is bounded by the computational complexity $O(I \cdot ((k + m + 1)|W| + k|V| + \frac{|\mathcal{C}_O| \cdot |W|}{c}))$. Since k, m, and c are usually much smaller than $|\mathcal{C}_O|$, the running time is basically bounded by $O(I \cdot |\mathcal{C}_O| \cdot |W|)$.

3.1.5 Experiments

In this section, we first introduce the data sets used in the experiment. Then we demonstrate the effectiveness of our semi-supervised topic modeling approach by showing two examples in two different scenarios. Finally, we also provide some quantitative evaluation.

Data Sets

Topic Desc.	Source	# of words	# of aspects
iPhone	CNET	4434	19
Barack Obama	wikipedia	312	14

Table 3.1: Basic Statistics of the REVIEW data set

Topic Desc.	Query Terms	# of articles	N
iPhone	iPhone	552	3000
Barack Obama	Barack+Obama	639	1000

Table 3.2: Basic Statistics of the BLOG data set

We need two types of data sets for evaluation. One type is expert reviews. We construct this data set by leveraging the existing services provided by CNET and wikipedia, i.e., we submit queries to their web sites and download the expert reviews on "iPhone" written by CNET editors[2] and the introduction part of articles about "Barack Obama" in wikipedia[3]. The composition and basic statistics of this data set (denoted as "REVIEW") is shown in Table 3.1.

[2]http://reviews.cnet.com/smart-phones/apple-iPhone-8gb-at/4505-6452_7-32309245.html?tag=pdtl-list
[3]http://en.wikipedia.org/wiki/Barack_Obama

The other type of data is a set of opinion sentences related to certain topic. In this paper, we only use Weblog data, but our method can be applied on any kind of data that contain opinions in free text. Specifically, we firstly submit topic description queries to Google Blog Search[4] and collect the blog entries returned. The search domain are restricted to spaces.live.com, since schema matching is not our focus. We further build a collection of N opinion sentences $\mathcal{C}_O = \{d_1, d_2, ..., d_N\}$ which are highly relevant to the given topic T using information retrieval techniques as described as the first stage in Section 3.1.3. The basic information of these collections (denoted as "BLOG" is shown in Table 3.2. For all the data collections, Porter stemmer [71] is used to stem the text and stop words in general English are removed.

Scenario I: Product

Gathering opinions on products is the main focus of the research on opinion mining, so our first example of opinion integration is a hot product, iPhone. There are 19 self-contained segments in the "iPhone" review of the REVIEW data set. We use these 19 segments as aspects from the review and define 11 extra aspects in the semi-supervised topic model.

We show part of the integration with review aspects in Table 3.3. We can see that there is indeed some interesting information discovered.

- In the "introduction" aspect (which corresponds to the background introduction part of the expert review), we see that lots of people care about the price of iPhone, and the sentences extracted from blog articles show different pricing information which confirms the fact that the price of iPhone has been adjusted. In fact, the first two sentences only mention the original price while the third sentence talks about the cut down of the price but the actual numbers are incorrect.

- The displayed sentence in the "activation" aspect describes the results if you do not

[4]http://blogsearch.google.com

Aspect	Expert Review	Similar Opinions	Supplementary Opinions
Introduction	Even with the new $399 price for the 8GB model (down from an original price of $599), it's still a lot to ask for a phone that lacks so many features and locks you into an iPhone-specific two-year contract with AT&T.		[**support=19**]The iPhone will come in two versions, a 4GB 499 model, and an 8GB 599 model with a two year contract.
			[**support=16**]The Price: 499 (4GB) or 599(8GB) with a two year contract , by the time the contract is over your iPhone will probably be scratched all over like the Nano or be made obsolete by better phone on the market.
			[**support=12**]Recently, Apple decided to cut down price of iPhone from 399 to 200 , giving rise to much rage from consumers bought the phone before.
Activation	You can make emergency calls, but you can't use any other functions, including the iPod music player.		[**support=10**]Several other methods for unlocking the iPhone have emerged on the Internet in the past few weeks, although they involve tinkering with the iPhone hardware or more complicated ways of bypassing the protections for AT T's exclusivity.
Battery	Battery life The Apple iPhone has a rated battery life of 8 hours talk time, 24 hours of music playback, 7 hours of video playback, and 6 hours on Internet use.	[**support=19**] iPhone will Feature Up to 8 Hours of Talk Time, 6 Hours of Internet Use, 7 Hours of Video Playback or 24 Hours of Audio Playback	[**support=7**]Playing relatively high bitrate VGA H.264 videos, our iPhone lasted almost exactly 9 freaking hours of continuous playback with cell and WiFi on (but Bluetooth off).

Table 3.3: iPhone Example: Opinion Integration with Review Aspects

activate the iPhone. A piece of very interesting information related to this aspect, "unlocking the iPhone" is never mentioned in the expert review but is extracted from blog articles by using our semi-supervised topic modeling approach. Indeed, we know that "unlock" or "hack" is a hot topic since the iPhone hit the market. This is a good demonstration that our approach is able to discover information which is highly related and supplementary to the review.

- The last aspect shown is about battery life. There is a high support ($support = 19$ in the column of similar opinions) of the life of battery described in the review, and there is another supplementary set of sentences ($support = 7$) which gives a concrete number of battery in hours under real usage of iPhone.

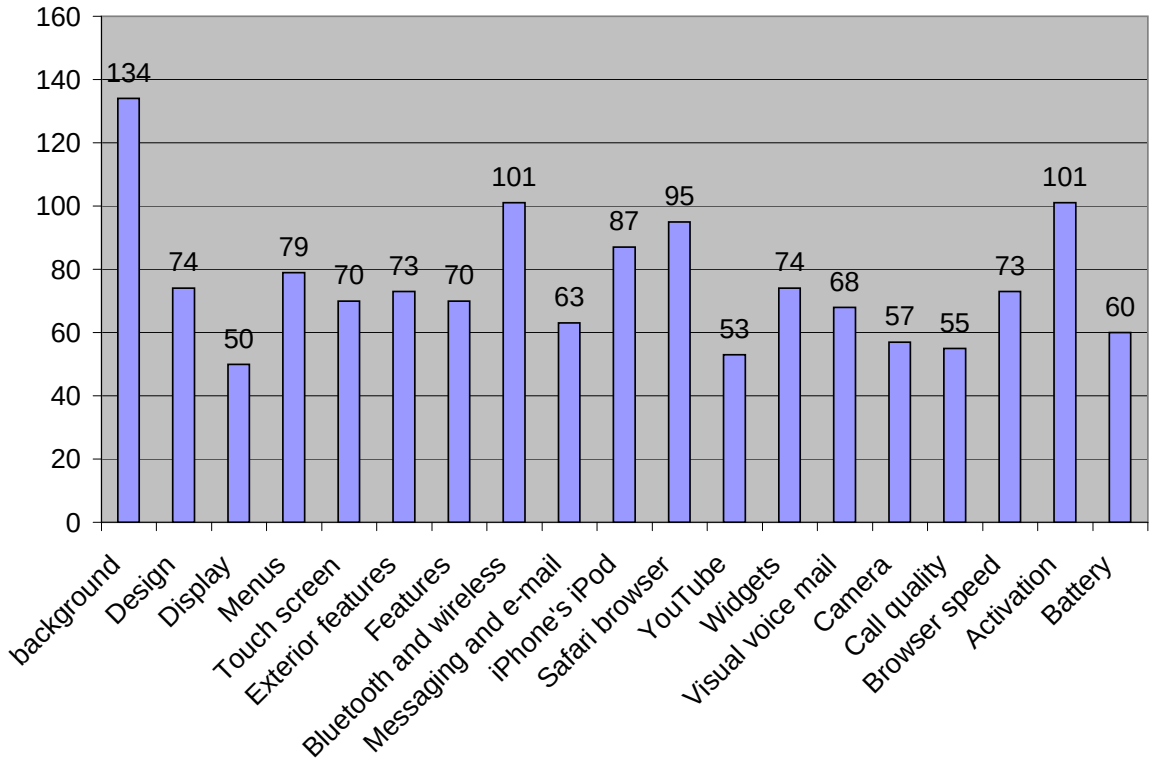


Figure 3.3: Support Statistics for iPhone Aspects

Furthermore, we may also want to know which aspects of iPhone people are most interested in. If we define the support of an aspect as the sum of the support of representative opinions in this aspect, we could easily get the support statistics for each review aspects

Supplementry Opinions on Extra Aspects
[support=15]You may have heard of iASign (http: iphone.fiveforty.net wiki index.php IASign), an iPhone Dev Wiki tool that allows you to activate your phone without going through the iTunes rigamarole.
[support=13]Cisco has owned the trademark on the name "iPhone" since 2000, when it acquired InfoGear Technology Corp., which originally registered the name.
[support=13]With the imminent availability of Apple's uber cool iPhone, a look at 10 things current smartphones like the Nokia N95 have been able to do for a while and that the iPhone can't currently match...

Table 3.4: iPhone Example: Opinion Integration on Extra Aspects

in our topic modeling approach. As can be seen in Figure 3.3, the "background" aspect attracts the most discussion. This is mainly caused by the mention of the price of iPhone in the background aspect. The next two aspects with highest support are "Bluetooth and Wireless" and "Activation" both with support 101. As stated in the iPhone review "The Wi-Fi compatibility is especially welcome, and a feature that's absent on far too many smart phones.", and our support statistics suggest that people do comment a lot about this unique feature of iPhone. "Activation" is another hot aspect as discovered by our method. As many people know, the activation of iPhone requires a two-year contract with AT&T, which brings much controversy among customers.

In addition, we show three of the most supported representative opinions in the extra aspects in Table 3.4. The first sentence points out another way of activating iPhone, while the second sentence brings up the information that Cisco was the original owner of the trademark "iPhone". The third sentence expresses a opinion in favor of another smartphone, Nokia N95, which could be useful information for a potential smartphone buyer who did not know about Nokia N95 before.

Scenario II: Political Figure

If we want to know more about a political figure, we could treat a short biography of the person as an expert review and apply our semi-supervised topic model. In this subsection, we demonstrate what we can achieve by an example of "Barack Obama". There is no definition of segments in the short introduction part in wikipedia, so we just treat each sentence as a segment in this experiment.

ID	Review	Similar Opinions	Supplementary Opinions
0	Barack Hussein Obama (born August 4, 1961) is the junior United States Senator from Illinois and a member of the Democratic Party.	**[support=9]**Senator Barack Hussein Obama is the junior United States Senator from Illinois and a member of the Democratic Party .	**[support=21]**Barack Obama, another leading Democratic presidential hopeful, campaigns for more dollars with "Dinner With Barack." **[support=11]**A Chicago, Illinois, radio station recently conducted a live survey on a man called Barack Obama. **[support=10]**In fact, there is not a single metropolitan area in the country where a family earning minimum wage can afford decent housing, said Senator Barack Obama.
1	The U.S. Senate Historical Office lists him as the fifth African American Senator in U.S. history and the only African American currently serving in the U.S. Senate.		**[support=16]**Barack Obama is an African American whose father was born in Kenya and got a sholarship to study in American.
3	He lived for most of his childhood in the majority-minority U.S. state of Hawaii and spent four of his pre-teen years in the multi-ethnic Indonesian capital city of Jakarta.		**[support=12]**Obama was born in Honolulu, Hawaii, to Barack Hussein Obama Sr., a Kenyan, and Kansas born Ann Dunham.
10	He is among the Democratic Party's leading candidates for nomination in the 2008 U.S. presidential election.	**[support=2]**Mr Obama will contest the Democrat presidential nomination	**[support=14]**(AP) Democratic presidential candidate Barack Obama said Sunday that the front runner for his party's nomination, Hillary Rodham Clinton, does not offer the break from politics as usual that voters need.
12	He married in 1992 and has two daughters.		**[support=3]**MARCH 4 Senator Barack Obama is threatening legal action against a self describ--ed pedophile who has posted photos of the Democratic politician's young daughters on a web site that purports to handicap the 2008 presidential campaign by evaluating the "cuteness" of underage daughters and granddaughters of White House aspirants

Table 3.5: Obama Example: Opinion Integration with Review Aspects

In Table 3.5, we display part of the opinion integration with the 14 aspects in the review. Since there is no short description of each aspect in this example, we use ID in the first column of the table to distinguish one aspect from another.

- Aspect 0 is a brief introduction of the person and his position, which attracts many sentences in the blog articles some directly confirming the information provided in the review, some also suggest his position while stating other facts.

- Aspect 1 and 3 talk about his heritage and early life, and we further discover from the blog articles supplementary information such as his birthplace is Honolulu, his parents' names are Barack Hussein Obama Sr. and Ann Dunham, and even why his father came to the US.

- For aspect 10 about his presidential candidacy, our summaries not only confirm the fact but also point out another democratic presidential candidate Hillary Clinton.

- A brief description of his family is in review aspect 12, and the mention of his daughters has attracted a piece of news related to young daughters of White House aspirants.

ID	**Review**	**Support**
0	Barack Hussein Obama (born August 4, 1961) is the junior United States Senator from Illinois and a member of the Democratic Party.	68
1	Born to a Kenyan father and an American mother, Obama grew up in culturally diverse surroundings.	36
12	He married in 1992 and has two daughters.	3

Table 3.6: Obama Example: Support of Aspects

After further summing up the support for each aspect, we display two of the most supported aspects and one least supported aspect in Table 3.6. The most supported aspect is aspect 0 with $Support = 68$, which as mentioned above is a brief introduction of the person and his position. Aspect 2 talking about his heritage ranks as the second with $Support = 36$,

which agrees with the fact that he is special among the presidential candidates because of his Kenyan origin and indicates that people are interested in it. The least covered aspect is aspect 12 about his family, since the total support is only 3.

Quantitative Evaluation

In order to quantitatively evaluate the effectiveness of our semi-supervised topic modeling approach, we designed a test which consists of three tasks, each asking a human user to perform a part of our processing. The main goal is to see to what extent our approach can reproduce the human choice. The test is designed based on the above-mentioned "Barack Obama" example. In order to reduce the bias, we collect the evaluation results from three users, who are all PhD students in our department, two males and one female.

In the first designed task, we aims at evaluating the effectiveness of our approach in identifying the extra aspects in addition to review aspects. Towards this goal, we generate a big set of sentences S_{all} by mixing all the sentences in $\{S_1^{sim}, ..., S_k^{sim}\} \cup \{S_1^{supp}, ..., S_k^{supp}\}$ with seven most supported sentences in $\{S_{k+1}, ..., S_{k+m}\}$. There are $|S_{all}| = 34$ sentences in S_{all} in total. The users are asked to select seven sentences from randomly permutated S_{all} that do not fit into the k review aspects. In this way, we could see how is the human consensus on this task and how our approach could recover the choice of human.

User	**Sentence ID of the 7 sentences**
Our Approach	2, 6, 9, 21, 22, 25, 30
User 1	1, 6, 9, 13, 16, 25, 30
User 2	9, 11, 16, 20, 21, 30, 31
User 3	2, 6, 8, 9, 24, 25, 31

Table 3.7: Selection of 7 Sentences on Extra Aspects

Table 3.7 displays the selection of the seven sentences on extra aspects by our method and the three users. The only sentence out of seven that all three users agree on is sentence number 9, which suggests that grouping sentences into extra aspects is quite a subjective task so it is difficult to produce results satisfactory to each individual user. However our

method is able to recover 52.4% of the user's choices on average.

In the second task, we try to evaluate the performance of our approach in grouping sentences into k review aspects. we randomly permute all the sentences in $\{S_1^{sim}, ..., S_k^{sim}\} \cup \{S_1^{supp}, ..., S_k^{supp}\}$ to construct a S_{review} and remove the aspect assigned to each sentence. For each of the 27 sentences, the users are asked to assign one of the 14 review aspects to it. In essence, this is a multi-class classification problem where the number of classes is 14.

The results turn out to be

- Three users agree on 13 sentences about the class label, which means that more than half of the sentences are controversial even among human users.

- On average, our method could recover the user's choices by 10.67 sentences out of 27. Note that if we randomly assign one aspect out of 14, (1) the probability of recovering k sentences out of 27 is

$$\binom{27}{k} \times pr^k \times (1 - pr)^{27-k}$$

  where $pr = \frac{1}{14}$. When $k = 10$, the probability is only around 0.00037; (2) the expected number of sentences recovered would be

$$\sum_{k=0}^{27} \binom{27}{k} \times pr^k \times (1 - pr)^{27-k} = 1$$

- Our method and all three users assigned the same label to 8 sentences.

- Among the mistakes our method made, three users only agree on 5 sentences. In other words, all three users assigned the same label to 5 sentences, and this label is different the label produced by our method.

Again, this task is subjective, and there is still much controversy among human users. But our approach performs reasonably : in the 13 sentences with human consensus, our method achieves the accuracy of 61.5%.

In the third task, our goal is to see how well we can separate similar opinions from supplementary opinions in the semi-supervised topic modeling approach. We first select 5 review aspects out of 14 which our method has identified both similar and supplementary opinions; then for each of the 5 aspects, we mix one similar opinion with several supplementary opinions; the users are supposed to select one sentence which share the most similar opinion with the review aspect. On average, our method could recover 60% of the choices of human users. Among the different choices between our method and the users, only one aspect has achieved consensus of three users. That is to say, this is a "true" mistake of our method, while other mistakes do not have agreement in the users.

3.1.6 Conclusions and Future Work

In this section, we formally defined a novel problem of integrating opinions expressed in a well-written expert review with those in various Web 2.0 sources such as Weblogs to generate an aligned integrated opinion summary. We proposed a new opinion integration method based on semi-supervised probabilistic topic modeling. With this model, we could automatically generate an integrated opinion summary that consists of (1) supporting opinions with respect to different aspects in the expert review; (2) opinions supplementary to those in the expert review but on the same aspect; and (3) opinions on extra aspects which are not even mentioned in the expert review. We evaluate our model on integrating opinions about two quite different topics (a product and a political figure) and the results show that our method works well for both topics.

Since integrating and digesting opinions from multiple sources is critical in many tasks, our method can be applied to develop many interesting applications in multiple domains. A natural future research direction would be to address the more general setup of the problem – integrating opinions in arbitrary text collections with multiple expert reviews instead of a single expert review.

3.2 Exploiting Structured Ontology

3.2.1 Overview

Intuitively, the good aspects in structured summary should be concise phrases that can both be easily interpreted in the context of the topic under consideration and capture the major opinions. However, where can we find such phrases and which phrases should we select as aspects? Furthermore, once we selected aspects, how should we order them to improve the readability of a structured summary? One way to generate aspects is to cluster all the opinion sentences and then identify representative phrases in each cluster. Although aspects selected in this way can effectively capture the major opinions, a major limitation is that it is generally hard to ensure that the selected phrases are relevant with the given topic [13].

In this section, we propose a novel approach to generating aspects by leveraging ontologies with structured information that are available online, such as the open domain knowledge base in Freebase[5]. Such kind of ontology data is not in small scale by any measure. For example, Freebase alone contains more than 10 million topics (i.e., entities), 3000 types (i.e., categories), and 30,000 properties (i.e., attributes); moreover, it is constantly growing as people collaboratively contribute. Freebase provides different properties for different types of topics such as personal information for a "US President" and product features for a "Digital Camera". Since this kind of resources can provide related entities/relations for a wide range of topics , our general idea is to leverage them as guidance for more informed organization of scattered online opinions, and in particular, to select the most interesting properties of a topic from such structured ontology as aspects to generate a structured opinion summary. A significant advantage of this approach to aspect generation is that the selected aspects are guaranteed to be very well connected with the topic, but it also raises an additional challenge in selecting the aspects to best capture the major opinions from a large number of aspects provided for each topic in the ontology. Different from some existing

[5] `http://www.freebase.com`

work on exploiting ontologies, e.g., [73], which relies on training data, we focus on exploring *unsupervised* approaches, which can be applied to a larger scope of topics.

Specifically, given a topic with entries in an ontology and a collection of scattered online opinions about the topic, our goal is to generate a structured summary where representative opinions are aligned with aspects and organized in an order easy for human to follow. We propose the following general approach: First, retrieval techniques are employed to align opinions to relevant aspects. Second, a subset of the most interesting aspects are selected. Third, we will further order the selected aspects to present them in a reasonable order. Finally, for the opinions not associated with the selected aspects from the ontology, we use a phrase ranking method to suggest new aspects to add to the ontology for increasing its coverage.

Implementing the second and third steps involves new challenges. In particular, without any training data, it is unclear how we should show the most interesting aspects in ontology with major opinions aligned and which presentation order of aspects is natural and intuitive for human. Solving these two challenges is the main focus of this work. We propose three methods for aspect selection, i.e., size-based, opinion coverage-based, and conditional entropy-based methods, and two methods for aspect ordering, i.e., ontology-ordering and coherence ordering. We evaluate our methods on two different types of topics: US Presidents and Digital Cameras. Qualitative results demonstrate the utility of integrating opinions based on structured ontology as well as the generalizability of proposed methods. Quantitative evaluation is also conducted to show the effectiveness of our methods.

3.2.2 Methods

Given (1) an input topic T, (2) a large number of aspects/properties $A = \{A_1, ..., A_m\}$ from an ontology that are related to T, and (3) a huge collection of scattered opinion sentences about the topic $D_T = \{s_1, \ldots, s_n\}$, our goal is to generate a structured organization of opinions that are both well aligned with the interesting aspects and representative of major

opinions about the topic.

The envisioned structured organization consists of a sequence of selected aspects from ontology ordered to optimize readability and a set of sentences matching each selected aspect. Once we obtain a set of sentences for each aspect, we can easily apply a standard text summarization method to further summarize these sentences. Thus the unique challenges related to our main idea of exploiting ontology, which are also the main focus of our study, are the following:

1. **Aspect Selection**: How can we select a subset of aspects $A' \subset A$ to capture the *major* or majority opinions in our opinion set D_T?
2. **Aspect Ordering**: How can we order a subset of selected aspects A' so as to present them in an order $\pi(A')$ that is most natural with respect to human perception?
3. **New Aspects Suggestion**: Can we exploit the opinions in D_T to suggest new aspects to be added to the ontology?

Aspect Selection

In order to align the scattered opinions to the most relevant aspects, we first use each aspect label $A_i \in A$ as a query to retrieve a set of relevant opinions in the collection $S_i \subseteq D_T$ with a standard language modeling approach, i.e., the KL-divergence retrieval model [88]. Up to 1000 opinion sentences are retrieved for each aspect; each opinion sentence can be potentially aligned to several aspects. In this way, scattered online discussion are linked to the most relevant aspects in the ontology, which enables a user to use aspects as "semantic bridges" to navigate into the opinion space..

However, there are usually a lot of candidate aspects in an ontology, and only some are heavily commented in online discussions, so showing all the aspects is not only unnecessary, but also overwhelming for users. To solve this problem, we propose to utilize the aligned

opinions to further select a subset of the most interesting aspects $A' \subset A$ with size k. Several approaches are possible for this subset selection problem.

- *Size-based*: Intuitively, the selected subset A' should reflect the major opinions. So a straightforward method is to order the aspects A_i by the size of the aligned opinion sentences S_i, i.e., the number of relevant opinion sentences, and then select the top k ones.

- *Opinion Coverage-based*: The previous method does not consider possible redundancy among the aspects. A better approach is to select the subset that covers as many *distinct* opinion sentences as possible. This can be formulated as a maximum coverage problem, for which a greedy algorithm is known to be a good approximation: we select one aspect at a time that is aligned with the largest number of uncovered sentences.

- *Conditional Entropy-based*: Aspects from a structured ontology are generally quite meaningful, but they are not designed specifically for organizing the opinions in our data set. Thus, they do not necessarily correspond well to the natural clusters in scattered opinions. To obtain aspects that are aligned well with the natural clusters in scattered opinions, we can first cluster D_T into l clusters $C = \{C_1, \ldots, C_l\}$ using K-means with $TF \times IDF$ as features, and then choose the subset of aspects that minimize Conditional Entropy of the cluster label given the aspect:

$$A' = \arg\min H(C|A') = \arg\min \left[- \sum_{A_i \in A', C_i \in C} p(A_i, C_i) \log \frac{p(A_i, C_i)}{p(A_i)} \right]$$

  This Conditional Entropy measures the uncertainty about the cluster label of a sentence given the knowledge of its aspect. Intuitively, if the aspects are aligned well with the clusters, we would be able to predict well the cluster label of a sentence if we know its aspect, thus there would be less uncertainty about the cluster label. In the extreme

Algorithm 1 Greedy Algorithm for Conditional Entropy Based Aspect Selection

Input: $A = \{A_1, ..., A_m\}$
Output: k-sized $A' \subseteq A$

$A' = \{\cup_{i=1}^{m} A_i\}$
for j=1 to k **do**
  $bestH = \infty$; $bestA = A_0$
  for each A_i in A **do**
    $tempA' = \{A_i, A' \setminus \{A_i\}\}$
    if $H(C|tempA') < bestH$ **then**
      $bestH = H(C|tempA')$
      $bestA = A_i$
  $A' = \{bestA, A' \setminus \{bestA\}\}$
output A'

case when the cluster label can be completely determined by the aspect, the conditional entropy would reach its minimum (i.e., 0). Intuitively, the conditional entropy-based method essentially selects the most appropriate aspects from the ontology to label clusters of opinions.

The exact solution of this combinatorial optimization problem is NP-complete, so we employ a polynomial time greedy algorithm to approximate it: in the i-th iteration, we select the aspect that can minimize the conditional entropy given the previous $i-1$ selected aspects. Pseudo code is given in Algorithm 1, where $A \setminus \{A_i\}$ represent the set A minus the element A_i.

Aspect Ordering

In order to present the selected aspects to users in a most natural way, it is important to obtain a coherent order of them, i.e., generating an order consistent with human perception. To achieve this goal, our idea is to use human written articles on the topic to learn how to organize the aspects automatically. Specifically, we would order aspects so that the relative order of the sentences in all the aspects would be as consistent with their order in the original online discussions as possible.

Formally, the input is a subset of selected aspects A'; each $A_i \in A'$ is aligned with a set of

relevant opinion sentences $S_i = \{S_{i,1}, S_{i,2}, ...\}$. We define a coherence measurement function over sentence pairs $Co(S_{i,k}, S_{j,l})$, which is set to 1 iff $S_{i,k}$ appears before $S_{j,l}$ in the same article. Otherwise, it is set to 0. Then a coherence measurement function over an aspect pair can be calculated as

$$Co(A_i, A_j) = \frac{\sum_{S_{i,k} \in S_i, S_{j,l} \in S_j} Co(S_{i,k}, S_{j,l})}{|S_i||S_j|}$$

As an output, we would like to find a permutation $\hat{\pi}(A')$ that maximizes the coherence of all pair-wise aspects, i.e.,

$$\hat{\pi}(A') = \arg\max_{\pi(A')} \sum_{A_i, A_j \in A', A_i \prec A_j} Co(A_i, A_j)$$

where $A_i \prec A_j$ means that A_i is before A_j. It is easy to prove that the problem is NP-complete. Therefore, we resort to greedy algorithms to find approximations of the solution. Particularly we view the problem as a ranking problem. The algorithm proceeds by finding at each ranking position an aspect that can maximize the coherence measurement, starting from the top of the rank list. The detailed algorithm is given in Algorithm 2.

New Aspects Suggestion

Finally, if the opinions cover more aspects than in the ontology, we also want to identify informative phrases to label such extra aspects; such phrases can also be used to further augment the ontology with new aspects.

This problem is similar to existing work on generating labels for clusters [87] or topic models [59]. Here we employ a simple but representative technique to demonstrate the feasibility of discovering interesting new aspects for augmenting the ontology. We first extract named entities from scattered opinions D_T using Stanford Named Entity Recognizer

Algorithm 2 Greedy Algorithm for Coherence Based Aspect Ordering

Input: A
Output: $\pi(A)$

for each A_i, A_j in A **do**
  calculate $Co(A_i, A_j)$
$\pi(A) = []$
for $p = 1$ to $len = A.size()$ **do**
  $Max = A_1$
  for each aspect A_i in A **do**
    $A_i.coherence = 0$
    for each aspect A_j in $\pi(A)$ **do**
      $A_i.coherence += Co(A_j, A_i)$ //aspects ranked before A_i
    for each aspect A_j in A, $j \neq i$ **do**
      $A_i.coherence += Co(A_i, A_j)$ //aspects ranked after A_i
    if $A_i.coherence > Max.coherence$ **then**
      $Max = A_i$
  remove Max from A; add Max to $\pi(A)$
output $\pi(A)$

[22]. After that, we rank the phrases by pointwise Mutual Information (MI):

$$MI(T, ph) = \log \frac{P(T, ph)}{P(T)P(ph)}$$

where T is the given topic and ph refers to a candidate entity phrase. $P(T, ph)$ is proportional to the number of opinion sentences they co-occur; $P(T)$ or $P(ph)$ are proportional to the number of times T or ph appears. A higher MI value indicates a stronger association. We can then suggest the top ranked entity phrases that are not aligned with selected aspects as new aspects.

3.2.3 Experiments

In this section, we first introduce the data sets used in the experiments. Then we show some qualitative sample results to demonstrate the utility of organizing scattered opinions under the guidance of structured ontology. Finally, we provide quantitative evaluation.

Statistics	Category 1 US president	Category 2 Digital Camera
Number of Topics	36	110
Number of Aspects	65±26	32±4
Number of Opinions	1001±1542	170±249

Table 3.8: Statistics of Data Sets

FreeBase Aspects	Supt	Representative Opinion Sentences
Appointees: - Martin Feldstein - Chief Economic Advisor	897	Martin Feldstein, whose criticism of Reagan era deficits has not been forgotten. Reagan's first National Security advisor was quoted as declaring...
Government Positions Held: - President of the United States - Jan 20, 1981 to Jan 20, 1989	967	1981 Jan 20, Ronald Reagan was sworn in as president as 52 American hostages boarded a plane in Tehran and headed toward freedom. 40th president of the US Ronald Reagan broke the so called "20 year curse"...
Vice president: - George H. W. Bush	847	8 years, 1981-1988 George H. W. Bush as vice president under Ronald Reagan... ...exception to the rule was in 1976, when George H W Bush beat Ronald.

Table 3.9: Opinion Organization Result for President Ronald Reagan

Data Sets

To examine the generalizability of our methods, we test on two very different categories of topics: *US Presidents* and *Digital Cameras*. For the ontology, we leverage Freebase, downloading the structured ontology for each topic. For the opinion corpus, we use blog data for US Presidents and customer reviews for Digital Cameras. The blog entries for US Presidents were collected by using Google Blog Search[6] with the name of a president as the query. Customer reviews for Digital Cameras were crawled from CNET[7]. The basic statistics of our data sets is shown in Table 3.8. For all the data collections, Porter stemmer [71] is applied and stop words are removed.

Sample Results

We first show sample results for automatic organization of online opinions. We use the opinion coverage-based algorithm in Section 3.2.2 to select 10 aspects (10-20 aspects were found to be optimal in [39]) and then apply the coherence-based aspect ordering method. The number of clusters is set so that there are on average 15 opinions per cluster.

[6] `http://blogsearch.google.com`
[7] `http://www.cnet.com`

FreeBase Aspects	Supt	Representative Opinion Sentences
Format: - Compact	13	Quality pictures in a compact package. ... amazing is that this is such a small and compact unit but packs so much power.
Supported Storage Types: - Memory Stick Duo	11	This camera can use Memory Stick Pro Duo up to 8 GB Using a universal storage card and cable (c'mon Sony)
Sensor type: - CCD	10	I think the larger ccd makes a difference. but remember this is a small CCD in a compact point-and-shoot.
Digital zoom: -2×	47	once the digital :smart" zoom kicks in you get another 3x of zoom I would like a higher optical zoom, the W200 does a great digital zoom translation...

Table 3.10: Opinion Organization Result for Sony Cybershot DSC-W200 Camera

Opinion Organization: Table 3.9 and Table 3.10 present sample results for President Ronald Reagan and Sony Cybershot DSC-W200 camera respectively[8]. We can see that (1) although Freebase aspects provide objective and accurate information about the given topics, extracted opinion sentences offer additional subjective information; (2) aligning scattered opinion sentences to most relevant aspects in the ontology helps digestion and navigation; and (3) the support number, which is the number of opinion sentences aligned to an aspect, can show the popularity of the aspect in the online discussions.

Adaptability of Aspect Selection: Being un-supervised is a significant advantage of our methods over most existing work. It provides flexibility of applying the methods in different domains without the requirement of training data, benefiting from both the ontology based template guidance as well as data-driven approaches. As a result, we can generate different results for different topics even in the same domain. In Table 3.11, we show the top three selected and ordered aspects for Abraham Lincoln and Richard Nixon. Although they belong to the same category, different aspects are picked up due to the differences in online opinions. People talk a lot about Lincoln's role in American Civil War and his famous quotation, but when talking about Nixon, people focus on ending the Vietnam war and the Watergate scandal. "Date of birth" and "Government position" are ranked first because people tend to start talking from these aspects, which is more natural than starting from aspects like "Place of death".

Baseline Comparison: We also show below the aspects for Lincoln generated by a repre-

[8]Due to space limit, we only show the first few aspects as output by our methods.

Supt	**Richard-Nixon**	Supt	**Abraham-Lincoln**
50	Date of birth: - Jan 9, 1913	419	Government Positions Held: - United States Representative Mar 4,1847-Mar 3,1849
108	Tracks Recorded: - 23-73 Broadcast: End of the Vietnam War	558	Military Commands: - American Civil War - United States of America
120	Works Written About This Topic: - Watergate	810	Quotations: - Nearly all men can stand adversity, but if you want to test a man's character, give him power.

Table 3.11: Comparison of Aspect Selection for Two Presidents (aligned opinions are omitted here)

Suggested Phrases	**Supporting Opinion Sentences**
Abraham Lincoln Presidential Library	CDB projects include the Abraham Lincoln Presidential Library and Museum
Abraham Lincoln Memorial	..., eventually arriving at Abraham Lincoln Memorial.
John Wilkes Booth	John Wilkes Booth shoots President Abraham Lincoln at Ford's Theatre ...

Table 3.12: New Phrases for Abraham Lincoln

sentative approach using clustering method (e.g. [24]). i.e., we label the largest clusters by selecting phrases with top mutual information. We can see that although some phrases make sense, not all are well connected with the given topic; using aspects in ontology circumvents this problem. This example confirms the finding in previous work that the popular existing clustering-based approach to aspects generation cannot generate meaningful labels [13].

```
Vincent
New Salem State Historic Site
USS Abraham Lincoln
Martin Luther King Jr
Gettysburg
John F.
```

New Aspect Discovery: Finally, in Table 3.12 we show some phrases ranked among top 10 using the method described in Section 3.2.2. They reveal additional aspects covered in online discussions and serve as candidate new aspects to be added to Freebase. Interestingly, John Wilkes Booth, who assassinated President Lincoln, is not explicitly listed in Freebase, but we can find it in people's online discussion using mutual information.

Evaluation of Aspect Selection

Measures: Aspect selection is a new challenge, so there is no standard way to evaluate it. It is also very hard for human to read all of the aspects and opinions and then select

Measures	SC	H	AC	AP	AAP
PRESIDENTS					
Random	503	1.9069	0.5140	0.0933	0.1223
Size-based	500	1.9656	0.3108	**0.1508**	0.0949
Opin Cover	**746**	1.8852	0.5463	0.0913	0.1316
Cond Ent.	479	**1.7687**	**0.5770**	0.0856	**0.1552**
CAMERAS					
Random	55	1.6389	0.6554	0.0871	0.1271
Size-based	70	1.6463	0.6071	**0.1077**	0.1340
Opin Cover	**82**	1.5866	0.6998	0.0914	0.1564
Cond Ent.	70	**1.5598**	**0.7497**	0.0789	**0.1574**

Table 3.13: Evaluation Results for Aspect Selection

a gold standard subset. Therefore, we opt to use indirect measures capturing different characteristics of the aspect selection problem (1) *Aspect Coverage* (AC): we first assign each aspect A_i to the cluster C_j that has the most overlapping sentences with A_i, approximating the cluster that would come into mind when a reader sees A_i. Then AC is defined as the percentage of the clusters covered by at least one aspect. (2) *Aspect Precision* (AP): for each covered cluster C_i, AP measures the Jaccard similarity between C_i as a set of opinions and the union of all aspects assigned to C_i. (3) *Average Aspect Precision* (AAP): defines averaged AP for all clusters where an uncovered C_i has a zero AP; it essentially combines AC and AP. We also report *Sentence Coverage* (SC), i.e., how many distinct opinion sentences can be covered by the selected aspects and *Conditional Entropy* (H), i.e., how well the selected aspects align with the natural clusters in the opinions; a smaller H value indicates a better alignment.

Results: We summarize the evaluation results in Table 3.13. In addition to the three methods described in Section 3.2.2, we also include one baseline of averaging 10 runs of random selection. The best performance by each measure on each data set is highlighted in bold font. Not surprisingly, opinion coverage-based approach has the best sentence coverage (SC) performance and conditional entropy-based greedy algorithm achieves the lowest H. Size-based approach is best in aspect precision but at the cost of lowest aspect coverage.

The trade-off between AP and AC is comparable to that between precision and recall as in information retrieval while AAP summarizes the combination of these two. The greedy algorithm based on conditional entropy outperforms all other approaches in AC and also in AAP, suggesting that it can provide a good balance between AP and AC.

Evaluation of Aspect Ordering

Human Annotation: In order to quantitatively evaluate the effectiveness of aspect ordering, we conduct user studies to establish gold standard ordering. Three users were each given k selected aspects and asked to perform two tasks for each US President: (1) identify clusters of aspects that are more natural to be presented together (*cluster constraints*) and (2) identify aspect pairs where one aspect is preferred to appear before the other from the viewpoint of readability. (*order constraints*). We did not ask them to provide a full order of the k aspects, because we suspect that there are usually more than one "perfect" order. Instead, identifying partial orders or constraints is easier for human to perform, thus provides more robust gold standard.

Human Agreement: After obtaining the human annotation results, we first study human consensus on the ordering task. For both types of human identified constraints, we convert them into pair-wise relations of aspects, e.g., "A_i and A_j should be presented together" or "A_i should be displayed before A_j". Then we calculate the agreement percentage among the three users. In Table 3.14, we can see that only a very small percentage of pair-wise partial orders (15.92% of the cluster constraints and none of the order constraints) are agreed by all the three users, though the agreement of clustering is much higher than that of ordering. This indicates that ordering the aspects is a subjective and difficult task.

Measures: Given the human generated gold standard of partial constraints, we use the following measures to evaluate the automatically generated full ordering of aspects: (1) *Cluster Precision* (pr_c): for all the aspect pairs placed in the same cluster by human, we calculate the percentage of them that are also placed together in the system output. (2)

AgreedBy	Cluster Constraint	Order Constraint
1	37.14%	89.22%
2	46.95%	10.78%
3	15.92%	0.00%

Table 3.14: Human Agreement on Ordering

Cluster Penalty (p_c): for each aspect pair placed in the same cluster by human, we give a linear penalty proportional to the number of aspects in between the pair that the system places; p_c can be interpreted as the average number of aspects between aspect pairs that should be presented together in the case of mis-ordering. Smaller penalty corresponds to better ordering performance. (3) *Order Precision* (pr_o): the percentage of correctly predicted aspect pairs compared with human specified order.

Results: In Table 3.15, we report the ordering performance based on two selection algorithms: opinion coverage-based and conditional entropy-based. Different selection algorithms provide different subsets of aspects for the ordering algorithms to operate on. For comparison with our coherence-based ordering algorithm, we include a random baseline and Freebase ontology ordering. Note that Freebase order is a very strong baseline because it is edited by human even though the purpose was not for organizing opinions. To take into account the variation of human annotation, we use four versions of gold standard: three are from the individual annotators and one from the union of their annotation. We did not include the gold standard that is the intersection of three annotators because that would leave us with too little overlap.

We have several observations: (1) In general, results show large variations when using different versions of gold standard, indicating the subjective nature of the ordering task. (2) Coherence-based ordering shows similar performance to Freebase order-based in cluster precision (pr_c), but when we take into consideration the distance-based penalty (p_c) of separating aspects pairs in the same cluster, coherence-based ordering is almost always significantly better except in one case. This shows that our method can effectively learn the coherence

Selection Algo	Gold STD	Cluster Random	Precision Freebase	(pr_c) Coherence	Cluster Random	Penalty Freebase	(p_c) Coherence	Order Random	Precision Freebase	(pr_o) Coherence
Opin Cover	1	0.3290	**0.9547**	0.9505	1.8798	0.1547	**0.1068**	0.4804	**0.7059**	0.4510
Opin Cover	2	0.3266	**0.9293**	0.8838	1.7944	0.3283	**0.1818**	**0.4600**	0.4000	0.4000
Opin Cover	3	0.2038	**0.4550**	0.4417	2.5208	**1.3628**	1.7994	0.5202	0.4561	**0.5263**
Opin Cover	union	0.3234	**0.7859**	0.7237	1.8378	0.6346	**0.4609**	**0.4678**	0.4635	0.4526
Cond Entropy	1	0.2540	**0.9355**	0.8978	2.0656	0.2957	**0.2016**	0.5106	**0.7111**	0.5444
Cond Entropy	2	0.2535	0.7758	**0.8323**	2.1790	0.7530	**0.5222**	0.4759	**0.6759**	0.5093
Cond Entropy	3	0.2523	0.4030	**0.5545**	2.3079	2.1328	**1.1611**	0.5294	0.7143	**0.8175**
Cond Entropy	union	0.3067	0.7268	**0.7488**	1.9735	1.0720	**0.7196**	0.5006	0.6500	**0.6833**

Table 3.15: Evaluation Results on Aspect Ordering

of aspects based on how their aligned opinion sentences are presented in online discussions. (3) Order precision (pr_o) can hardly distinguish different ordering algorithm. This indicates that people vary a lot in their preferences as which aspects should be presented first. However, in cases when the random baseline outperforms others the margin is fairly small, while Freebase order and coherence-based order have a much larger margin of improvement when showing superior performance.

3.2.4 Conclusions and Future Work

A major challenge in automatic integration of scattered online opinions is how to organize all the diverse opinions in a meaningful way for any given topic. In this section, we propose to solve this challenge by exploiting related aspects in structured ontology which are guaranteed to be meaningful and well connected to the topic. We proposed three different methods for selecting a subset of aspects from the ontology that can best capture the major opinions, including size-based, opinion coverage-based, and conditional entropy-based methods. We also explored two ways to order aspects, i.e., ontology-order and coherence optimization. In addition, we also proposed appropriate measures for quantitative evaluation of both aspect selection and ordering.

Experimental evaluation on two data sets (US President and Digital Cameras) shows that by exploiting structured ontology, we can generate interesting aspects to organize scattered opinions. The conditional entropy method is shown to be most effective for aspect selection, and the coherence optimization method is more effective than ontology-order in optimizing the coherence of the aspect ordering, though ontology-order also appears to perform reasonably well. In addition, by extracting salient phrases from the major opinions that cannot be covered well by any aspect in an existing ontology, we can also discover interesting new aspects to extend the existing ontology.

Complementary with most existing summarization work, this work proposes a new direction of using structured information to organize and summarize unstructured opinions,

opening up many interesting future research directions. For instance, in order to focus on studying aspect selection and ordering, we have not tried to optimize sentences matching with aspects in the ontology; it would be very interesting to further study how to retrieve sentences matching each aspect most accurately. Another promising future work is to organize opinions using both structured ontology information and well-written overview articles.

Chapter 4

Aspect Level Sentiment Analysis

After integrating opinions from different sources and organizing them into meaningful aspects, we looked into the problem of inferring the sentiments in the opinions with respect to each aspect, so as to provide the users with a more detailed and informed multi-perspective view of the opinions.

4.1 Sentiment Rated Aspect Summarization

4.1.1 Overview

Generally, given a target entity, we can obtain many user-generated comments each often also with an overall rating. For example, users review and rate the products on CNET[1] from one to five stars; on eBay[2], buyers leave feedback comments to the seller and rate the transaction as positive, neutral or negative. Usually the number of comments about a target entity is of a very large scale, such as hundreds of thousands, and the number is consistently growing as more and more people keep contributing online. So the question is how to help a user better digest such a large number of comments.

In this section, we propose to generate a "rated aspect summary" which provides a decomposed view of the overall ratings for the major aspects so that a user can gain different perspectives towards the target entity. This kind of decomposition is quite useful because different users may have quite different needs and the overall ratings are generally not infor-

[1]http://www.cnet.com/
[2]http://www.ebay.com/

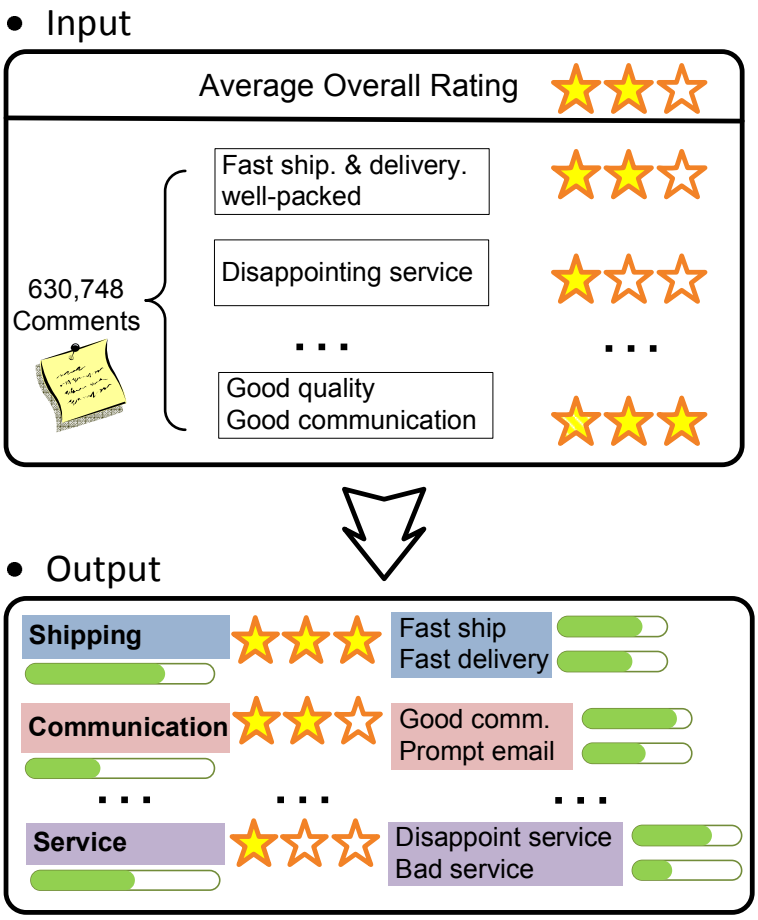


Figure 4.1: Problem Setup

mative enough. For example, a prospective eBay buyer may compromise on shipping time but not on product quality. In this case, it is not sufficient for the buyer to just know the overall ratings of a seller, and it would be highly desirable for the buyers to know the ratings of a seller on the *specific* aspect about product quality.

Rated aspect summarization can potentially help users make wiser decisions by providing more detailed information. This problem setup is illustrated in Figure 4.1. The input data represents what users normally can see through a community comment website, which generally consists of a large number of short comments with companion overall ratings. With such data, a user can only get an overall impression by looking at the average overall rating; it is tedious and time-consuming to go over the large number of comments for more detailed analysis. In contrast, in the generated rated aspect summary (shown as output),

the overall rating is decomposed into several aspects; each aspect has support information (the green bars) showing the confidence on the aspect rating; representative phrases with support information further enrich the rated aspects, and can serve as indices to navigate into a set of specific comments about this aspect.

This kind of rated aspect summarization is also helpful even if users do explicitly give ratings for some given aspects, because (1) we may still want to further decompose the ratings into finer sub-aspects. For example, people typically rate "food" in restaurant reviews, but users usually want to know in what sense the food is good or bad. Is there concern about healthiness or about taste? (2) the given aspects may not cover all the major aspects discussed in the text comments. In the eBay system, there are four defined aspects to rate a seller, called Detailed Seller Ratings (DSR), namely "Item as described", "Communication", "Shipping time" and "Shipping and handling charges". But it would be difficult to know the seller's performance on "packaging", "price", or "service", which might be more useful for some potential buyers.

To the best of our knowledge, this rated aspect summarization problem has not been studied in the existing work, though it is related to some existing work on opinion summarization (the connection is further discussed in Section 2). Specifically, no previous work has attempted or proposed algorithms to decompose an overall rating into ratings on ad hoc aspects learned from the comments.

Our goal is to solve this novel summarization problem with no human supervision, or with minimum supervision in the case when the user wants to specify keywords to describe aspects that should be used to summarize the comments and decompose the rating. We propose to solve the rated aspect summarization problem in three steps: (1) extract major aspects; (2) predict rating for each aspect from the overall ratings; (3) extract representative phrases. In the first step, we propose a topic modeling method, called Structured PLSA, modeling the dependency structure of phrases in short comments. It is shown to improve the quality of the extracted aspects when compared with two strong baselines. In the second

step, we propose to predict the aspect ratings using two different approaches, both unsupervised: Local Prediction uses the local information of the overall rating of a comment to rate the phrases in that comment; Global Prediction rates phrases based on aspect level rating classifiers which are learned from overall ratings of all comments. After the first two steps, we have the comments segmented into different aspects and different rating values. Then we could select phrases that represent what have been mostly said in this aspect.

Since this is a new task, there is no existing data set that can be used to evaluate it. We opt to create our own test set using the seller feedback comments from eBay. We design measures to evaluate each of the three components in a rated aspect summary (i.e., aspects, ratings of aspects, and representative phrases). The extracted aspects are evaluated by comparing aspect coverage and clustering accuracy against human generated aspect clusters; we use the DSR ratings in eBay as the gold standard to evaluate the aspect rating prediction, and evaluation metrics include both aspect rating correlation and ranking loss; we calculate precision and recall of the representative phrases against human labeled phrases. Evaluation results show that our proposed methods can generate useful rated aspect summaries from large amounts of short comments and overall ratings. The PLSA approach, especially the proposed Structured PLSA which leverages the phrase structures in the short comments, outperforms the k-means clustering method. Our results also show that Global Prediction generates more accurate rating prediction, but Local Prediction is sufficient at predicting a few representative phrases in each aspect.

4.1.2 Problem Definition

Given a large number of short comments about a target entity, each associated with an overall rating indicating different levels of overall opinion, our goal is to generate a rated aspect summary, i.e. an aspect summary with a rating for each aspect, in order to help users better digest the comments along different dimensions of the target entity. There are two application scenarios:

1. No supervision: If there is no prior knowledge of the aspects, we just automatically decompose the overall rating into purely ad hoc aspects based on the data.

2. Minimum supervision: If the user could provide a couple of keywords specifying aspects he or she would be interested in, we should accommodate targeted aspect decomposition.

Formally, we denote the collection of short comments by $T = \{t_1, t_2, ...\}$, where each $t \in T$ is associated with an overall rating of $r(t)$.

Definition (Overall Rating) An overall rating $r(t)$ of a comment t is a numerical rating indicating different levels of overall opinion of t, and $r(t) \in \{r_{min}, ..., r_{max}\}$.

Usually, it is infeasible for a user to go over all the overall ratings of a large number of comments. A common way used in many real applications is to summarize them with a single number: the average overall ratings of the whole collection.

Definition (Average Overall Rating) The average overall rating of a collection of comments $R(T)$ is a score averaged over all the overall ratings: $R(T) = \frac{\sum_{t \in T} r(t)}{|T|} \in [r_{min}, r_{max}]$. This is often used in existing web sites to summarize the users' overall ratings.

In short comments, such as the eBay feedback text, most opinions are expressed in concise phrases, such as "well packaged", "excellent seller". So with the help of some shallow parsing techniques, we could extract those phrases and identify the head term and the modifier. This also allows us to take advantage of the phrase structure to learn aspects.

Definition (Phrase) A phrase $f = (w_m, w_h)$ is in the form of a pair of head term w_h and modifier w_m. Usually the head term is an aspect or feature, and the modifier expresses some opinion towards this aspect.

Then each comment is represented by a bag of phrases $t = \{f = (w_m, w_h) | f \in t\}$ instead of a regular bag of words. After that, rated aspect summarization could be naturally decomposed into three steps:

1. Identify k major aspect clusters

2. Predict aspect rating for each aspect

3. Extract representative phrases to support or explain the aspect ratings

Some of the concepts are defined as follows:

Definition (Aspect Cluster) An aspect cluster A_i is a cluster of head terms that share similar meaning in the given context. Those words jointly represent an aspect that users would comment on and/or would be interested in. We denote $A_i = \{w_h | A(w_h) = i\}$, where $A(.)$ is a mapping function produced by some aspect clustering algorithm that maps a head term to a cluster label.

Definition (Aspect Rating) An aspect rating $R(A_i)$ is a numerical measure with respect to the aspect A_i, showing the degree of satisfaction demonstrated in the comments collection T toward this aspect, and $R(A_i) \in [r_{min}, r_{max}]$.

Definition (Representative Phrase) A representative phrase $rf = (f, s(f))$ is a phrase f with a support value $s(f)$, where $s(f) \in [1, \infty)$ indicating how many phrases in the comments that this phrase can represent.

Note that, we use $r(.)$ to denote a discrete rating (an integer between r_{min} and r_{max}), and $R(.)$ to denote an average rating over a number of discrete ratings, which is a rational number (usually non-integer) between r_{min} and r_{max}. We can now define the rated aspect summary we would like to generate as follows.

Definition (Rated Aspect Summary) A rated aspect summary is a set of tuples $(A_i, R(A_i), RF(A_i))_{i=1}^{k}$, where A_i is a ratable aspect, $R(A_i)$ is the predicted rating on A_i, and $RF(A_i)$ is a set of representative phrases in this aspect.

4.1.3 Methods

We propose several methods to solve the problem of rated aspect summarization in three steps as defined in Section 4.1.2.

Aspect Discovery and Clustering

As stated in Section 4.1.2, in short comments, opinions on different aspects are usually expressed in concise phrases. We assum that each phrase is parsed into a pair of head term w_h and modifier w_m in the form of $f = (w_m, w_h)$. Usually the head term is about an aspect or feature, and the modifier expresses some opinion towards this aspect. In the first step, our task is to identify k interesting aspects and cluster head terms into those aspects. We propose three different approaches.

1. k-means Clustering Intuitively, the structure of phrases should help with the clustering of the head terms, because if two head terms tend to use the same set of modifiers, they should share similar meaning. For example, head terms that are usually modified by "fast" should be more similar to each other compared with head terms modified by "polite" or "honest". So in the first attempt, we try to use the relation between modifiers and head terms by representing each head term w_h as a vector $v(w_h)$ in the form of

$$v(w_h) = (c(w_h, w_m^1), c(w_h, w_m^2), ...)$$

where $c(w_h, w_m^i)$ is the number of co-occurrences of head term w_h with modifier w_m^i. Then we apply k-means [53], a standard clustering algorithm shown to be effective in many clustering tasks, to a set of such vectors. The clusters output by k-means form the aspects of interest. However, the space of modifiers is usually of very high dimensionality, ranging from several hundreds to thousands. Due to the *curse of dimensionality*, the sparsity of the data could affect the clustering performance.

2. Unstructured PLSA Probabilistic latent semantic analysis (PLSA) [32] and its extensions [89, 60, 57] have recently been applied to many text mining problems with promising results. If we ignore the structure of the phrases, we could apply PLSA on the head terms to extract topics, i.e. aspects.

   We define k unigram language models: $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$ as k theme models, each is a multinomial distribution of head terms, capturing one aspect. A comment $t \in T$ can then be regarded as a sample of the following mixture model.

$$p_t(w_h) = \sum_{j=1}^{k} [\pi_{t,j} p(w_h|\theta_j)]$$

   where w_h is a head term, $\pi_{t,j}$ is a comment-specific mixing weight for the j-th aspect ($\sum_{j=1}^{k} \pi_{t,j} = 1$). The log-likelihood of the collection T is given by

$$\texttt{log}p(T|\Lambda) = \sum_{t \in T} \sum_{w_h \in V_h} \{c(w_h, t) \times \texttt{log} \sum_{j=1}^{k} [\pi_{t,j} p(w_h|\theta_j)]\}$$

   where V_h is the set of all the head terms, $c(w_h, t)$ is the count of head term w_h in comment t, and Λ is the set of all model parameters.

   After estimating the model with an Expectation-Maximization (EM) algorithm [19], we have a set of theme models extracted from the text collection $\{\theta_i | i = 1, ..., k\}$, and now we could group each head term $w_h \in V_h$ into one of the k aspects by choosing the theme model with the largest probability of generating w_h, which is our clustering mapping function:

$$A(w_h) = \arg\max_j p(w_h|\theta_j)$$

   Intuitively, if two head terms tend to co-occur with each other (such as, "ship" and "delivery" co-occurring in "fast ship and delivery") and one term is assigned a high probability, then the other generally should also be assigned a high probability in

order to maximize the data likelihood. Thus, this model generally captures the co-occurrences of head terms and can help cluster the head terms into aspects based on co-occurrences in comments.

3. Structured PLSA Using a similar intuition as in the k-means clustering method, we try to incorporate the structure of phrases into the PLSA model, using the co-occurrence information of head terms and their modifiers.

   Similar to Unstructured PLSA, we define k unigram language models of head terms: $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$ as k theme models. Each modifier could be represented by a set of head terms that it modifies:

$$d(w_m) = \{w_h | (w_m, w_h) \in T\}$$

   which can then be regarded as a sample of the following mixture model.

$$p_{d(w_m)}(w_h) = \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h|\theta_j)]$$

   where $\pi_{d(w_m),j}$ is a modifier-specific mixing weight for the j-th aspect, which sums to one, i.e. $\sum_{j=1}^{k} \pi_{d(w_m),j} = 1$. The log-likelihood of the collection of modifiers V_m is

$$\texttt{log}p(V_m|\Lambda) = \sum_{w_m \in V_m} \sum_{w_h \in V_h} \{c(w_h, d(w_m)) \times$$

$$\texttt{log} \sum_{j=1}^{k} [\pi_{d(w_m),j} p(w_h|\theta_j)]\}$$

   where $c(w_h, d(w_m))$ is the number of co-occurrences of head term w_h with modifiers w_m, and Λ is the set of all model parameters. Using a similar EM algorithm as in Section 2, we could estimate the k theme models of head terms and obtain the clustering mapping

function. For completeness, we are showing the updating formulas as follows:

$$
\begin{aligned}
p(z_{d(w_m),w_h,j}) &= \frac{\pi^{(n)}_{d(w_m),j} p^{(n)}(w_h|\theta_j)}{\sum_{j'=1}^{k} \pi^{(n)}_{d(w_m),j'} p^{(n)}(w_h|\theta_{j'})} \\
\pi^{(n+1)}_{d(w_m),j} &= \frac{\sum_{w_h \in V_h} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j})}{\sum_{j'} \sum_{w_h \in V_h} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j'})} \\
p^{(n+1)}(w_h|\theta_j) &= \frac{\sum_{w_m \in V_m} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j})}{\sum_{w'_h \in V_h} \sum_{w_m \in V_m} c(w'_h, d(w_m)) p(z_{d(w_m),w'_h,j})}
\end{aligned}
$$

where $p(z_{d(w_m),w_h,j})$ represents the probability of head term w_h associated with modifier w_m assigned to the jth aspect.

Compared with Unstructured PLSA, this method models the co-occurrence of head terms at the level of the modifiers they use instead of at the level of comments they occur. Since we are working on short comments, there are usually only a few phrases in each comment, so the co-occurrence of head terms in comments is not very informative. In contrast, Structured PLSA model goes beyond the comments and organizes the head terms by their modifiers, which could use more meaningful syntactic relations.

4. Incorporating Aspect Priors In many cases, we have some domain knowledge about the aspects. For instance, "food" and "service" are the major aspects in comments on restaurants. And sometimes a user may have specific preference on some aspects. For example, a buyer may be especially into the "packaging" aspect. In the probabilistic model framework, we could use conjugate prior to incorporate such human knowledge to guide the clustering of aspects.

   Specifically, we build a unigram language model $\{p(w_h|a_j)\}_{w_h \in V_h}$ for each aspect a_j that we have prior knowledge about. For example, a language model for a "packaging"

aspect may look like

$$\begin{aligned} p(\texttt{packaging}|a_1) &= 0.5 \\ p(\texttt{wrapping}|a_1) &= 0.5 \end{aligned}$$

The we could define a conjugate prior (i.e., a Dirichlet prior) on each unigram language model, parameterized as $Dir(\{\sigma_j p(w_h|a_j)+1\}_{w_h \in V_h})$, where σ_j is a confidence parameter for the prior. Since we use a conjugate prior, σ_j can be interpreted as the "equivalent sample size" which means that the effect of adding the prior would be equivalent to adding $\sigma_j p(w_h|a_j)+1$ pseudo counts for head term w_h when we estimate the topic model $p(w_h|\theta_j)$. Basically, the prior serves as some "training data" to bias the clustering results.

The prior for all the parameters is given by

$$p(\Lambda) \propto \prod_{j=1}^{k} \prod_{w_h \in V_h} p(w_h|\theta_j)^{\sigma_j p(w_h|a_j)}$$

where $\sigma_j = 0$ if we do not have prior knowledge on some aspect θ_j.

We can then use the Maximum A Posteriori (MAP) estimator to estimate all the parameters as follows (for Unstructured PLSA and Structured PLSA respectively)

$$\begin{aligned} \hat{\Lambda} &= \arg\max_{\Lambda} p(T|\Lambda)p(\Lambda) \\ \hat{\Lambda} &= \arg\max_{\Lambda} p(V_m|\Lambda)p(\Lambda) \end{aligned}$$

The MAP estimate can be computed using essentially the same EM algorithm as presented above with only slightly different updating formula for the component language models. The new updating formulas are: (for Unstructured PLSA and Structured

PLSA respectively)

$$p(w_h|\theta_j)^{(n+1)} = \frac{\sum_{t \in T} c(w_h, t) p(z_{t,w_h,j}) + \sigma_j p(w_h|a_j)}{\sum_{w'_h \in V_h} \sum_{t \in T} c(w'_h, t) p(z_{t,w'_h,j}) + \sigma_j}$$

$$p(w_h|\theta_j)^{(n+1)} = \frac{\sum_{w_m \in V_m} c(w_h, d(w_m)) p(z_{d(w_m),w_h,j}) + \sigma_j p(w_h|a_j)}{\sum_{w'_h \in V_h} \sum_{w_m \in V_m,} c(w'_h, d(w_m)) p(z_{d(w_m),w'_h,j}) + \sigma_j}$$

Aspect Rating Prediction

In the second step, we already have k aspect clusters of head terms in the form of a clustering mapping function $A(.)$. We want to predict the rating for each aspect from the overall rating without any supervision nor any external knowledge. We first propose two methods for classifying each phrase f into a rating $r(f)$ and then aspect ratings could be calculated by aggregating ratings of the phrases within each aspect.

1. Local Prediction In the first method, we assume that what a user writes in the comment is consistent with the overall rating she gives. In other words, each phrase mentioned in a comment shares the same rating as the overall rating of the comment. This kind of prediction uses only the *local* information which is the overall rating of the exact comment that the phrase appears in. So the rating classifier for a phrase is

$$r(f \in t) = r(t) \in \{r_{min}, ..., r_{max}\}$$

   which basically classifies the phrase into the same overall rating as the comment.

2. Global Prediction In the second method, we do not blindly rate each phrase with the overall rating of the comment it appears in. Instead, we first learn aspect level rating classifiers using the *global* information of the overall ratings of all comments. Then each phrase is classified by the globally learned rating classifier. The main idea is that by learning rating classifiers globally, we hope to correct some errors made when we only have local information available.

Specifically, for each aspect A_i, we estimate $r_{max} - r_{min} + 1$ rating models empirically, each corresponding to a rating value $r \in \{r_{min}, ..., r_{max}\}$. Each rating model is a unigram language model of modifiers capturing the distribution of modifiers with the given rating value. We estimate the rating model by the empirical distribution:

$$p(w_m|A_i, r) = \frac{c(w_m, S(A_i, r))}{\sum_{w'_m \in V_m} c(w'_m, S(A_i, r))}$$

where

$$S(A_i, r) = \{f | f \in t, A(f) = i, \texttt{ and } r(t) = r\}$$

is the subset of phrases that belong to this aspect and comments containing these phrases receive the overall rating of r. After that we can classify each phrase by choosing the rating class that has the highest probability of generating the modifier in the phrase, which is basically a Naive Bayes classifier with uniform prior on each rating class.

$$r(f) = \arg\max_r \{p(w_m|A_i, r) | A(f) = i\}$$

Intuitively, the phrase rating classifier of Global Prediction should work better than that of Local Prediction. In some cases, not all the phrases in a comment are consistent with the overall rating. It is quite possible that people give a high overall rating while mentioning some shortcomings in the comments, and vice-versa. Suppose a comment says "slow shipping" while rated as maximum score: Local Prediction would blindly rate the phrase a maximum score; but Global Prediction could potentially tell "slow" is a low-rating on shipping, because "slow" should appear in more lowly rated comments than highly rated comments about shipping. With the globally learned classifiers, Global Prediction should be able to accommodate more noisy data, where some comments do not totally agree with their overall ratings.

3. Rating Aggregation After we classify each phrase into different rating values using

either Local Prediction or Global Prediction, the rating for each aspect A_i can be calculated by aggregating the rating of the phrases that are clustered into this aspect. Following the common practice, we calculate the average rating of phrases within this aspect.

$$R(A_i) = \frac{\sum_{A(f)=i} r(f)}{|\{f|A(f) = i\}|}$$

$R(A_i)$ is some value between r_{min} and r_{max}, representing the average rating towards this aspect.

Representative Phrases Extraction

In the third step, we are trying to pull out some representative phrases in order to provide the users with some textual clues for better understanding of the predicted aspect rating. If our aspect clusters and aspect rating predictions are accurate, we would expect the phrases that are classified into the same aspect and same rating to be very similar to each other. So we could segment the collection of comments T into subsets of phrases for each aspect A_i and each rating value r,

$$F(A_i, r) = \{f|A(f) = i, r(f) = r\}$$

Then we could extract the top three phrases with the highest frequency in each subset. The support value for a phrases f is the frequency of the phrase in the subset

$$s(f) = c(f, F(A_i, r))$$

4.1.4 Experiments

Rated aspect summarization is a new task which has not been studied before, so there is no existing data set available to evaluate it. In this section, we describe how we create a data set using the sellers' feedback comments on eBay. Then we present our experimental results and show both qualitative and quantitative evaluation of our methods using this data set.

Data Set and Preprocessing

We create a data set by collecting feedback comments for 28 eBay sellers with high feedback scores for the past year. The feedback score of a seller is defined as the accumulated number of positive feedback. In eBay, the feedback mechanism works as follows: after each transaction, the buyer is supposed to leave some feedback for the seller, including (1) an overall rating as positive, neutral or negative (2) Detailed Seller Ratings (DSRs) on four given aspects "Item as described", "Communication", "Shipping time" and "Shipping and handling charges" at the scale of 5 stars (3) some short comments in free text.

Then for preprocessing, we utilize the POS tagging and chunking function of the OpenNLP toolkit[3] to identify phrases in the form of a pair of head term and modifier. Some statistics of the data set is shown in Table 4.1.

Statistics	**Mean**	**STD**
# of comments per seller	57,055	62,395
# of phrases per comment	1.5533	0.0442
overall rating (positive %)	0.9799	0.0095

Table 4.1: Statistics of the Data Set

There are a few observations from the statistics: (1) Those sellers with high feedback scores receive large number of comments, $57,055$ on average. But the number also varies across different sellers, as the standard deviation is very high. (2) The buyers usually use only a few phrases in each comment. After parsing, there are about 1.5 phrases per comment. Note that, the original data is more noisy. For example, user-invented superlative "AAA+++" does not provide much detailed information on aspects. Our preprocessing reduces the data by about 40% in terms of the number of tokens. (3) The average overall ratings are usually very high, nearly 0.98 are positive, so they are not discriminative.

[3]http://opennlp.sourceforge.net/

	Aspects	Rating	Phrases of Rating 1	Phrases of Rating 0
1	described,promised	4.8457	as described (3993) as promised (323) as advertised (149)	than expected (68) than described(43) i ordered (10)
2	shipped,arrived	4.3301	quickly shipped (162) great thanks (149) quickly arrived (138)	open box (39) wrong sent (29) back sent (15)
3	recommended, was	3.9322	highly recommended (236) highly recommend (115) exactly was (84)	back be (42) defective was (40) not have (37)
4	shipping,delivery	4.7875	fast shipping (5354) quick shipping (879) fast delivery (647)	good shipping (170) slow shipping (81) reasonable shipping (32)
5	transaction, item	4.6943	great item (1017) great transaction (704) smooth transaction (550)	wrong item (70) new condition (48) new item (34)
6	seller,product	4.9392	great seller (2010) great product (1525) good seller (866)	poor communication (12) defective product (12) personal comm (9)
7	works,price	4.3830	great works (1158) great price (642) good price (283)	perfectly works (132) fine works (90) not working (29)
8	buy, do	4.0917	will buy (356) would buy (347) again buy (271)	not did (105) not work (91) didnt work (49)

Table 4.2: A Sample Result of Rated Aspect Summarization

Sample Result of Rated Aspect Summarization

A sample rated aspect summarization for one of the sellers is shown in Table 4.2. The first column shows the automatically discovered and clustered aspects using Structured PLSA. We empirically set the number of aspects to be 8. The top two head terms in each aspect are displayed as the aspect label. The second column is the predicted ratings for different aspects using Global Prediction. Due to the mostly-positive nature of the eBay feedback, we treat both neutral and negative as rating of 0, and positive as rating of 1. So our predicted rating for each aspect would be a value between 0 and 1. Then we uniformly map our predicted rating to the 5 star ratings to produce a score between 0 and 5 as in the second column of the table. The last two columns show three representative phrases together with

their frequency for each aspect and for rating 1 and 0 respectively.

We observed that

1) We can discover the major aspects and cluster the head terms in a meaningful way. Aspect 1 is about whether the seller truly delivers what is promised; Aspect 3 shows whether the buyers would recommend this seller; Aspect 7 talks about price. Almost all aspects are coherent and separable except that aspect 2 and aspect 4 are both talking about "shipping time".

2) The aspect ratings help us gain some insight towards this seller's performance on different aspects.

3) Although some phrases are noisy, such as "not did" and "i ordered" and some phrases are miss-classified into ratings, such "new condition" and "new item" misclassified into the rating 0 class, majority of the phrases are informative and indicate the ratings they belong to. In addition, the frequency counts could help users tell whether these opinions are representative of the major opinions.

4) We could see some correlation between the predicted aspect ratings and the phrase frequency counts: usually a high aspect rating maps to a large number of phrases in rating 1 and a small number of phrases in rating 0 and vice-versa.

We also show a sample comparison of two sellers in Table 4.3. Due to the limit of space, only part of the summary is displayed. We can see that although two sellers have very similar overall rating (98.66% positive V.S. 98.16% positive), Seller1 is better at providing good shipping while Seller2 is stronger at good communication. This clearly provides more detailed information than the overall rating, showing the benefit of decomposing an overall rating into aspect ratings.

Evaluation of Aspect Discovery and Clustering

In order to quantitatively evaluate the effectiveness of aspect discovery and clustering, we ask users to manually generate some aspect clusters as our gold standard. For each seller,

Aspects	Seller1	Seller2
OVERALL	98.66%	98.16%
described	4.7967	4.8331
communication	4.5956	**4.9462**
shipping	**4.9131**	4.2244

Table 4.3: Sample Comparison of Two Sellers

we display no more than 100 head terms that have support no less than 0.1%. (for a typical seller, there are about 80 terms) We also display the term frequency and five most frequent phrases with this head term. An example is

```
price   608  0.012
        great price, good price, fair price,
        nice price, reasonable price
```

where the head term is "price", which appears 608 times in this seller's feedback comments, accounting for 1.2% of all the head terms; and the most frequent phrases with this head term are "great price, good price, fair price, nice price, reasonable price". These phrases are displayed mainly to provide the user with some context for clustering the head terms in case there is any ambiguity. Then we ask the users to cluster them into no more than 8 clusters based on their meanings. If more than 8 clusters are formed, the user is supposed to keep the top 8 clusters with highest support. Each cluster is supposed to be an aspect that a buyer would comment on. Some head terms that do not look like aspects (maybe because of parsing errors) or do not fit into top 8 clusters could be ignored.

After obtaining the human annotated gold standard for 12 sellers, we evaluate the aspect clustering algorithms by both **Aspect Coverage** and **Clustering Accuracy**.

Aspect Coverage aims at measuring how well an aspect clustering algorithm could recover or match the major aspects that human identified. We count it as an aspect match if the most frequent term in an algorithm cluster appears as one of the terms in a human identified cluster. Top K clusters are the K clusters with the largest size. Then we define Aspect Coverage at top K as the number of aspect matches within top K clusters divided by K.

However, Aspect Coverage only evaluates the most frequent term in each cluster (it could be treated as the label of a cluster); it does not measure the coherence of terms within a cluster. So we propose to use Clustering Accuracy to measure the clustering coherence performance. Given a head term w_h, let $h(w_h)$ and $A(w_h)$ be the human annotated cluster label and the label generated by some algorithm, respectively. The clustering accuracy is defined as follows:

$$\texttt{Clustering Accuracy} = \frac{\sum_{w_h \in V_h} \delta(h(w_h), map(A(w_h)))}{|V_h|}$$

where $|V_h|$ is the total number of head terms, $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(A(w_h))$ is the permutation mapping function that maps each cluster label $A(w_h)$ to the equivalent label from the human annotation. The best mapping can be found by using the Kuhn-Munkres algorithm [50].

We compare three aspect clustering methods on Aspect Coverage in Figure 4.2 and on Clustering Accuracy in Table 4.4. As seen in Figure 4.2, both probabilistic methods, i.e. Unstructured PLSA and Structured PLSA, are good at picking up a small number of the most significant aspects (when K is small). As the number of clusters increases, the performance of three methods converge to a similar level, around 0.8. This indicates that all of the three methods could discover the 8 major aspects reasonably well. However, based on Table 4.4, structured PLSA achieves the best performance of Clustering Accuracy, 0.52 in bold font, meaning that the clusters are most coherent with respect to human generated clusters. This is consistent with our analysis in Section 4.1.3.

Method	**Clustering Accuracy**
k-means	0.36
Unstructured PLSA	0.32
Structured PLSA	**0.52**

Table 4.4: Evaluation of Cluster Accuracy

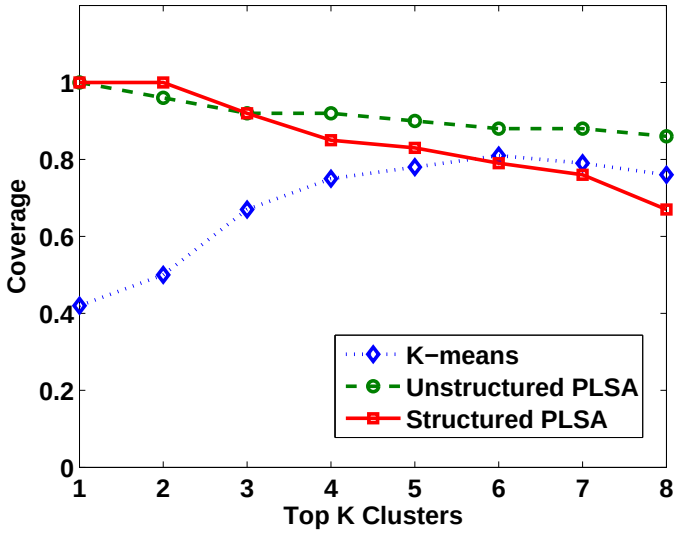


Figure 4.2: Evaluation of Aspect Coverage

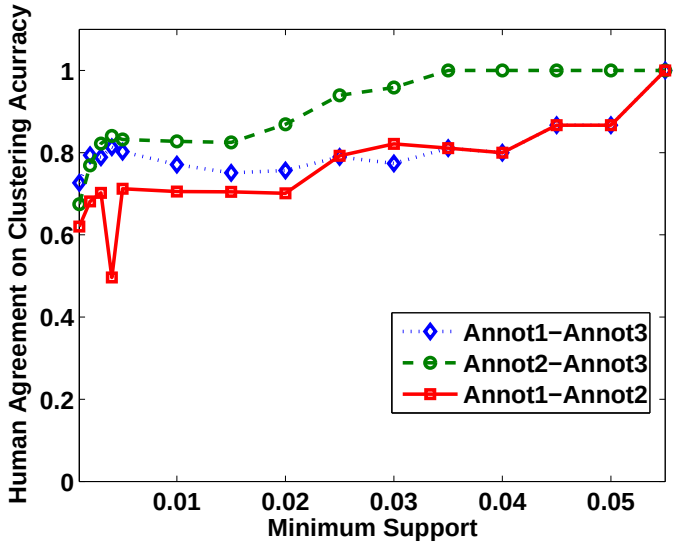


Figure 4.3: Human Agreement Curve on Clustering Accuracy

	Seller1	Seller2	Seller3	AVG
Annot1-Annot2	0.6610	0.5484	0.6515	0.6203
Annot1-Annot3	0.7846	0.6806	0.7143	0.7265
Annot2-Annot3	0.7414	0.6667	0.6154	0.6745
AVG	0.7290	0.6319	0.6604	**0.6738**

Table 4.5: Human Agreement on Clustering Accuracy

We would also like to test how well human do with respect to coherence in such a clustering task, so that we could have some sense of the "upper bound" performance. Three users are asked to label the same set of three sellers. Then the human agreement is evaluated as the clustering accuracy between each pair of users, as shown in Table 4.5. It can be seen that human agreement could vary a lot, from 0.5484 to 0.7846, across different annotator pairs and different data they work on. The average agreement is 0.6738. We plot the human agreement curve with different cutoffs of head term support values in Figure 4.3. The higher the support value is, the smaller number of head terms there will be. We would expect human to agree more on smaller number of terms. Indeed, the three curves of Clustering Accuracy, denoting three pairs of annotators, converge to 1 at some point of support value 5.5%, where there are only three or four terms left. Before that point of minimum support, most agreement still stays no more than 0.8. All these evidences show that aspect discovery and cluster could be a subjective and difficult task.

Evaluation of Aspect Rating Prediction

It is more difficult to evaluate the aspect rating prediction with human generated gold standard, because it would be too costly to ask human to read all the comments and rate each ad hoc aspect. Instead, we use the DSR ratings by buyers as the gold standard. As discussed in Section 4, we could use the descriptions for the four DSR criteria as priors when estimating the four aspect models, so that the discovered aspects would align with the DSR criteria defined in the eBay system. After that, we map our predicted ratings into $[0, 5]$ in order to allow comparison with the actual DSR ratings provided by buyers. Note that

our algorithms do not use any information from the true DSR ratings. Instead we predict the DSR ratings based on only the comments and the overall ratings. If our algorithms are accurate, the predictions are expected to be similar to the true DSR ratings by the buyers who wrote the comments.

Since the aspect rating prediction also depends on the quality of aspect clusters, we compare our two methods of rating prediction (Local Prediction and Global Prediction) using three different aspect clustering algorithms proposed in Section 4.1.3. Note that, there is no easy way to incorporate such prior information into the k-means clustering algorithm. So we map the k-means clusters to four DSR criteria as a post processing step: we align the k-means cluster to a DSR if that cluster contains the description word of the DSR; if such alignment cannot be found for some DSR, we just randomly pick a cluster. We also include a baseline in our comparison which is using the positive feedback percentage to predict each aspect without extracting aspects from the comments.

We propose to evaluate the prediction from two perspectives: **Aspect Ranking Correlation** and **Ranking Loss**. Aspect Rank Correlation measures the effectiveness of ranking the four DSRs for a given seller. For example, a seller may be better at "shipping" than at "communication". We use both Kendall's Tau rank correlation and Pearson's correlation coefficient. Ranking loss [74] measures the average distance between the true and predicted ratings. The ranking loss for an aspect is equal to

$$\sum_i \frac{|actual\_rating_i - predicted\_rating_i|}{N}$$

where $N = 28$ is the number of sellers. Average ranking loss on K aspects is simply the average over each aspect. The results are shown in Table 4.8, and the best performance of each column is marked in bold font. A good prediction should have high correlation and low ranking loss. It can be seen that

- The aspect clustering quality indeed affects the prediction of aspect ratings. If we

use k-means to cluster the aspects, no matter which prediction algorithm we use, the prediction performance is poor, even below the baseline performance especially for correlation.

- The prediction algorithm Global Prediction performs always better than Local Prediction at correlation for both Unstructured and Structured PLSA aspect clustering. This indicates that the ratings predicted by Global Prediction are more discriminative and accurate in ranking the four DSRs.

- The ranking loss performance of our methods Unstructured PLSA/Structured PLSA + Local Prediction/Global Prediction is almost always better than the baseline. The best ranking loss averaged among the four DSRs is 0.2287 given by Structured PLSA + Local Prediction compared with the baseline of 0.2865.

- The ranking loss performance also varies a lot across different DSRs. The difference is most significant on DSR 4, which is about "shipping and handling charges". However, the problem is that "charges" almost never occur in the comments, so that the aspect cluster estimated using this prior is kind of randomly related to "shipping and handling charges", resulting in the low performance on the prediction on this aspect. If we exclude this aspect and take the average of the other three ranking losses, average ranking loss performance of each algorithm improves and the best performance is achieved by Structured PLSA + Global Prediction at 0.1534 compared with 0.2365 by the baseline.

Evaluation of Representative Phrases Extraction

In order to generate gold standard for representative phrases, we utilize both the true DSR ratings and human annotation. The DSR ratings are used to generate candidate phrases at different rating level. The assumption is that if a buyer gives a low rating (less or equal to 3 out of 5) on an aspect, he or she will express negative opinion on this aspect in the text

DSR Criteria	Phrases of Rating 1	Phrases of Rating 0
Item as described	as described (15609) as promised (1282) as expected 487	than expected (6)
Communication	great communication (1164) good communication (1018) excellent communication (266)	poor communication (22) bad communication (12)
Shipping time	fast shipping (28447) fast delivery (3919) quick shipping (3812)	slow shipping (251) slow delivery (20) not ship (18)
Shipping and handling charges		excessive postage (10)

Table 4.6: Sample Representative Phrases by Human Annotation

comments. In order to rule out the bias from our aspect clustering algorithm, we do not distinguish aspects for the phrases when displaying the phrases to the users. To summarize, we aggregate the comments with low DSR ratings and high DSR ratings respectively, and then display the most frequent 50 phrases in each set. The user is asked to select three most frequent phrases for opinions of rating 1 and rating 0 on each of the four aspects. An example output from the human annotation is as in Table 4.6.

Basically, the user is given a list of candidates for rating 1 phrases and a list of candidates for rating 0 phrases, and is then asked to fill in the eight cells as in Table 4.6. In some cases, there are no phrases that fit into some cell, such as no positive phrases for "shipping and handling charges" in this case, that cell is simply left as empty.

We apply our representative phrases extraction algorithm on top of different aspect clustering and rating prediction algorithms, and output three phrases for each of the eight cells in Table 4.6.

Then we treat each cell as a "query", human generated phrases as "relevant document", and computer generated phrases as "retrieved document". Then we can calculate precision and recall as in evaluation of information retrieval:

$$\texttt{Precision} = \frac{|\{relevant\_docs\} \cap \{docs\_retrieved\}|}{|\{docs\_retrieved\}|}$$

$$\texttt{Recall} = \frac{|\{relevant\_docs\} \cap \{docs\_retrieved\}|}{|\{relevant\_docs\}|}$$

Methods	**Prec.**	**Recall**
k-means + Local Prediction	0.3055	0.3510
k-means + Global Prediction	0.2635	0.2923
Unstructured PLSA + Local Prediction	0.4127	0.4605
Unstructured PLSA + Global Prediction	0.4008	0.4435
Structured PLSA + Local Prediction	**0.5925**	**0.6379**
Structured PLSA + Global Prediction	0.5611	0.5952

Table 4.7: Evaluation of Representative Phrases

We report the average precision and average recall in Table 4.7 based on human annotation of 10 sellers. Note that when the user is filling out the cells in the table, he or she is also classifying the phrases into the four aspects and removing the phrases that are not of the right rating. So it is also an indirect way of evaluating our aspect clustering and aspect rating prediction algorithms. As we can tell from the table, (1) No matter which rating prediction algorithm we use, Structured PLSA always outperforms Unstructured PLSA which is always better than k-means; This is consistent with previous results. (2) Local Prediction always outperforms Global Prediction, independent of the underlying aspect clustering algorithm. This indicates that Local Prediction is sufficient and even better than Global Prediction at selecting only a few representative phrases for each aspect. (3) The best performance is achieved by Structured PLSA + Local Prediction at average precision of 0.5925 and average recall of 0.6379.

4.1.5 Conclusions and Future Work

In this section, we formally defined a novel problem of rated aspect summarization, which aims at decomposing the overall ratings for a large number of short comments into ratings on the major aspects so that a user can gain different perspectives towards the target entity. We proposed several general methods to solve the problem in three steps. With our methods, we

could automatically generate a rated aspect summary that consists of (1) a number of major aspects; (2) predicted ratings for each of the major aspects; and (3) representative phrases that explain the predicted ratings. We have demonstrated the feasibility of automatically generating such a summary by using the seller feedback comments data of eBay. We also propose several ways to quantitatively evaluate such a new task. Results show that (1) aspect clustering is a subjective task with low human agreement, but our methods, especially Structured PLSA, perform reasonably well. (2) although based on simple assumption, Local Prediction is usually sufficient for predicting a few representative phrases in each aspect. But Global Prediction provides rating prediction with more discrimination in ranking different aspects.

For the future work, we plan to combine the three steps into one optimization framework so that they could potentially benefit from each other. We are also planning to evaluate our methods on other kinds of data, such as product reviews. Another interesting future direction is to study how to compare entities (e.g. sellers, products) more effectively based on the rated aspects.

Aspect Clustering	**Aspect Prediction**	**Correlation**				**Ranking Loss**			
		Kendal's Tau	**Pearson**	**DSR1**	**DSR2**	**DSR3**	**DSR4**	**AVG/4**	**AVG/3**
baseline		0.2892	0.3161	0.1703	0.2053	0.3332	0.4372	0.2865	0.2363
k-means	Local Pred	0.1106	0.1735	0.1469	0.1925	0.3116	0.4177	0.2672	0.2170
k-means	Global Pred	0.1225	-0.0250	1.3954	0.2726	0.2242	0.3750	0.5668	0.6307
Unstructured PLSA	Local Pred	0.2815	0.4158	0.1402	0.1439	0.3092	0.3514	0.2362	0.1977
Unstructured PLSA	Global Pred	**0.4958**	0.5781	0.2868	**0.1262**	**0.2172**	0.4228	0.2633	0.2101
Structured PLSA	Local Pred	0.1905	0.4517	0.1229	0.1386	0.3113	**0.3420**	**0.2287**	0.1909
Structured PLSA	Global Pred	0.4167	**0.6118**	**0.0901**	0.1353	0.2349	0.5773	0.2594	**0.1534**

Table 4.8: Evaluation Results on Aspect Rating Prediction

4.2 Aspect-Dependent Sentiment Lexicon

In the previous section, we introduce methods for gerating a sentiment rated aspect summary by inferring aspect level sentiment ratings only from the opinion text and associated overall sentiment ratings. This kind of approach has the advantage that it can dynamically infer the sentiment of words based on the given domain, while traditional sentiment dictionary methods are static. However, when the data is too sparse to infer accurate aspect ratings, traditional methods using sentiment dictionary may be more reliable. To this end, we propose a novel framework that combines the advantage of both.

More specifically, we formulate the problem as automatic construction of an aspect-dependent sentiment lexicon, which can be used to score any free opinion text to produce an aspect level sentiment rating. We propose a novel optimization framework that provides a unified and principled way to combine different sources of information for learning such a sentiment lexicon, including opinion text, overall sentiment ratings, sentiment dictionary and synonym/antonym dictionaries.

4.2.1 Overview

People have studied many sentiment analysis applications, such as opinion retrieval, opinion question answering, opinion mining, opinion summarization and sentiment classification. Essential to most of these applications is a comprehensive and high quality sentiment lexicon. Such a lexicon is not only necessary for sentiment analysis when no training data is available (in such a case, supervised learning would be infeasible), but is also useful for improving the effectiveness of any supervised learning approach to sentiment analysis through providing high quality sentiment features [14].

However, there is no general-purpose sentiment lexicon that is optimal for all domains, because it is well known that sentiments of words are sensitive to the topic domain [82]. For example, "unpredictable" is negative in the electronics domain while being positive in

the movie domain. Indeed, sentiment lexicons adapted to the particular domain or topic have been shown to improve task performance in a number of applications, including opinion retrieval [63, 37], and expression level sentiment classification [14]. Nevertheless, little attention has been paid to the further challenge that even in the same domain the same word may still indicate different polarities with respect to different aspects in context. For example, in laptop domain, "large" is negative for the battery aspect while being positive for the screen aspect.

In this chapter, we focus on the problem of constructing a sentiment lexicon that is not only domain specific but also dependent on the aspect in context. Here, we use context-aware, context-dependent and aspect-dependent interchangeably, all referring to the expected output of a sentiment score assigned to each aspect and opinion word combination (e.g. BATTERY:large:-1). In particular, we are interested in methods generally applicable to any unlabeled opinionated corpus in any topical domain, so we make no assumption of the availability of human judged labels which are usually expensive to obtain in a new domain. Instead, we identify several sources of easy-to-collect information that are useful for determining the context-dependent sentiment of words. To solve the challenge that multiple signals come in different formats and may even cause contradictions, we combine them through appropriate constraints in the objective function of a novel optimization framework, in which we search for optimal assignments of sentiment scores to aspect-opinion pairs that are most consistent with all the constraints. In this way, the optimization framework provides a unified and principled way to automatically construct a domain-specific aspect-dependent sentiment lexicon by consolidating multiple evidences from different sources.

More specifically, in the objective function, we combine the following four kinds of soft constraints, capturing four different sources of knowledge about sentiment, respectively: (1) constraints for sentiment priors which come from general-purpose sentiment lexicons, (2) constraints for overall sentiment ratings which provide the overall sentiments for all the words combined in the reviews, (3) constraints for similar sentiments which can be collected from

synonyms in a thesaurus or from parsing the opinion collection with sentiment coherency assumption i.e. "and" rules as in linguistics heuristics, and (4) constraints for opposite sentiments which are from antonyms in a thesaurus or "but" rules in linguistics heuristics. These constraints cover most of the heuristics that have been exploited in existing work for inferring domain specific sentiments, and our method is the first to combine all those heuristics in a general and unified framework. More importantly, our constructed sentiment lexicon is not only domain specific but also aspect dependent.

To evaluate the effectiveness of our proposed framework, we conduct experiments on data sets in two different domains: hotel reviews and customer feedback surveys on printers. The results show that our approach can not only identify new sentiment words specific to the given domain (e.g. "private" is positive in hotel reviews; "compatible" is positive about printers) but also determine the different polarities of a word depending on the aspect in context (e.g. "huge room" v.s. "huge price" for hotels; "cheap ink" v.s. "cheap appearance" for printers). To further quantitatively evaluate the lexicon quality, we create a gold standard lexicon through human annotation, and our method is proved to be effective in constructing a high quality aspect-dependent sentiment lexicon. The results also demonstrate the advantage of combining multiple evidences over using any single evidence. Moreover, since the value of sentiment lexicons mostly lies in their usefulness in applications, we also study the performance of an aspect-level sentiment classification task by using the automatically constructed lexicon. The results show that using the context-dependent sentiment lexicon constructed by our optimization framework improves the sentiment classifier, compared with using baselines or a competitive method.

4.2.2 Problem Definition

We first define a general-purpose sentiment lexicon.

Definition (General-Purpose Sentiment Lexicon) A general-purpose sentiment lexicon L is

a dictionary of opinion words where each word w is assigned a score representing the degree of sentiment. Conventionally, the sentiment score $L(w) \in [-1, 1]$; and in many cases it is binary, i.e. either $+1$ (positive) or -1 (negative).

Our goal is to automatically construct a context-dependent sentiment lexicon, which can be used to supplement the general sentiment lexicon and provide more accurate context-dependent sentiment information for different applications, such as sentiment classification, opinion summarization, opinion retrieval and so on.

To construct a context-dependent sentiment lexicon, we assume that a set of aspects are given: $A = \{A_1, A_2, \cdots, A_k\}$, where each aspect is defined as follows:

Definition (Aspect) An aspect A_i is a set of terms characterizing a subtopic or a theme in a given domain, which can be features of products or attributes of services. For example, words such as "breakfast", "restaurant", and "pizza" can characterize the aspect about food in hotel reviews. We denote an aspect by $A_i = \{a;\ f(a) = i\}$, where $f(a)$ is a mapping function from a word a to its aspect index i.

Such aspects can be obtained through domain experts manual effort, or unsupervised automatic methods (e.g. [38]), or automatic methods with specified user interests as minimal human supervision (e.g. [55]). It is not our focus to find those aspects. Instead, assuming the availability of aspects, our problem is to automatically construct a context-dependent sentiment lexicon, defined as follows:

Definition (Context-Dependent Sentiment Lexicon) A context-dependent sentiment lexicon L_c is a dictionary of opinion words conditioned on different aspects of the given domain. Each entry in L_c is a pair of aspect A_i and opinion word w, and it is assigned a score representing the positive or negative sentiment it is expressing. $L_c(A_i, w) \in [-1, 1]$.

Our general idea of constructing such a lexicon is to leverage many naturally available resources, which we will discuss in detail in the next section.

4.2.3 Multiple Sources of Useful Signals

We do not make any assumption about the availability of human judged labels because they are usually expensive to obtain in a new topic domain. Nevertheless, we identify several kinds of easy-to-collect information that are helpful signals in determining the context-dependent sentiments of words. Here we summarize and categorize different sources of signals.

1. **General-purpose sentiment lexicon**, which contains words that are almost always positive or negative in any domain, such as "excellent" and "poor". This lexicon provides high confidence but low coverage sentiments.

2. **Overall sentiment rating**, i.e. sentiment rating/score at the document level. In many cases, each opinionated text comes with an overall sentiment rating from the user, such as in TripAdvisor[4], Epinions[5], and Amazon[6] reviews. Such kind of data is abundant on the Web. For example, there are more than 40 million travel-related reviews on TripAdvisor, and millions of reviews on millions of products from Epinions. The intuition is that the overall rating conveys some information about the sentiment expressed in the text. For example, it is very unlikely that a user uses all negative words in the text while giving an overall rating of 5 stars.

3. **Thesaurus**, which contains synonym and antonym information, such as WordNet[7]. For example, we may not know whether "large" is positive or negative for the screen aspect in laptop reviews, but we know it should be very similar to "big" and very different from "tiny". Then if we have some other evidences about the polarity of "big" or "tiny", we can better infer the polarity of "large".

4. **Linguistic heuristics**

[4] http://www.tripadvisor.com
[5] http://www.epinions.com
[6] http://www.amazon.com
[7] http://wordnet.princeton.edu

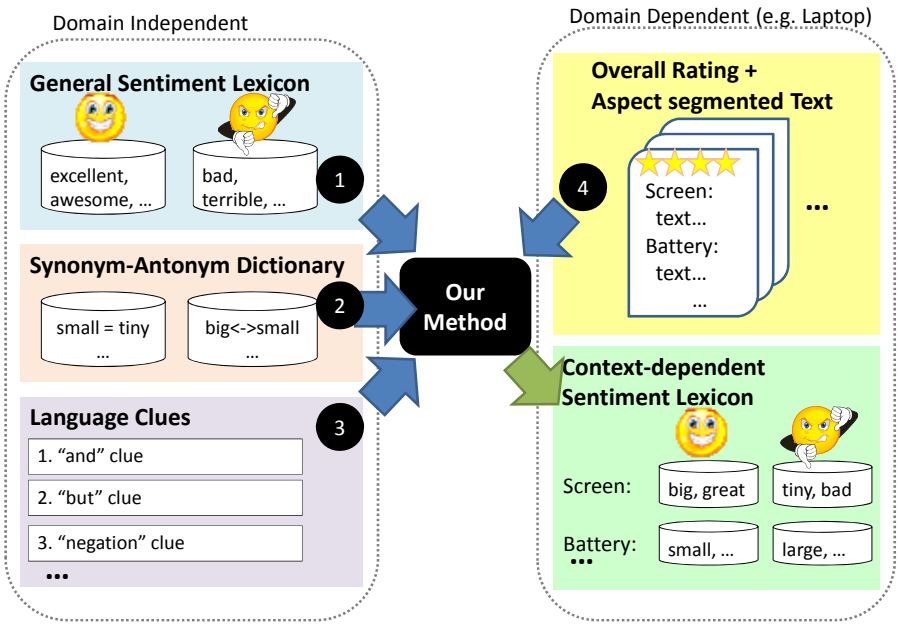


Figure 4.4: Problem Overview

(a) "and" rule: Clauses that are connected with "and"-like conjunctives usually express the same sentiment polarity. For example, "battery lasts long and screen size is large" implies that "long" for "battery" and "large" for "screen size" are of the same polarity. Other terms include: as well as, likewise.

(b) "but" rule: Clauses that are connected with "but"-like conjunctives usually express the opposite sentiment polarity. For example, "battery lasts long but screen size is tiny" indicates that "long" for "battery" and "tiny" for "screen size" are of the opposite polarity. Other terms include: however, nevertheless, though, although, except that, except for, besides, with the exception of, despite, in spite of.

(c) "negation" rule: Negation words such as "no", "not", and "never" reverse the sentiment of the opinion word in the same clause. For instance, "not happy" should have opposite sentiment as "happy".

These categories cover most of the heuristics used in existing works of learning domain specific sentiment lexicon, but no previous work has combined all these sources of signals.

Since the information from any single source can be sparse, it would be helpful if we can combine the signals from multiple sources effectively. To this end, we propose to combine all the information from different signals and learn a context-dependent sentiment lexicon, as illustrated in Figure 4.4. The idea is that when the signal from one source is not available or not confident enough, we can still refer to other signals to fill the gap. In the next section, we propose a novel optimization framework to effectively combine different kinds of signals in a unified way.

4.2.4 An Optimization Framework

Due to the fact that the different signals come in different formats, it is not clear how to combine them in a unified way. Moreover, there can be contradictory signals from different sources, which we also need to deal with. We first discuss how we generate all the candidate lexicon entries which form the search space for the optimization problem, and then define components in the objective function to capture various constraints. Finally, we show how we transform the proposed optimization framework into a linear programming problem which has efficient solutions and locally optimal solutions are also guaranteed to be globally optimal.

Generation of Candidate Lexicon Entries

The goal of this step is to tag the text collection with aspects and extract candidate opinion words to be paired with the aspects. After that, the pairs serve as entries in the context-dependent sentiment lexicon which are going to be assigned with polarity scores by our optimization method.

It is common to use each sentence as a tagging unit. But it is often the case, especially in online reviews, that one sentence covers different aspects in several subsentences or clauses; in addition, one clause can express sentiment of different polarity than other clauses in the same sentence. Thus, we choose to use clauses as units instead of sentences; this allows

us to associate potential opinions words with the aspects more accurately. We employ the Stanford Parser to do the sentence splitting and to parse sentences into syntactic tree structures. Then we use the subtrees tagged as "simple declarative clause", as candidate clauses. We also manually set a few rules to merge fragmental clauses into longer and more meaningful ones.

After that, we can now tag each clause s with the corresponding aspects. Since we already have a set of defined aspects A in the form of word clusters, we take the straightforward way that is to tag the clause with the aspects whose word cluster overlaps with the words in the clause. Now we have the opinionated text segmented into clauses which are tagged with the corresponding aspects. An example sentence is as follows, where two clauses are in brackets: the first clause is tagged with the SERVICE aspect because "check in" appears in the word cluster of SERVICE; similarly, the second clause is tagged with the FOOD aspect.

```
[The (check in):SERVICE is very smooth] and
[the (restaurant):FOOD is the best].
```

Finally, the other non-aspect and non-stop words in each clause are considered potential opinion words in the context of the tagged aspects. In the previous example, we will extract the pairs (SERVICE, very) and (SERVICE, smooth) from the first clause and (FOOD, best) from the second clause. If one clause has been tagged with more than one aspect, we will pair the potential opinion words with each aspect. It is possible to employ other aspect segmentation and tagging techniques to extract the candidate pairs, but we choose a simple and trackable approach here in order to focus on the next step of sentiment learning.

Constraints in the Objective Function

We propose to formulate this as an optimization problem. Basically, we will be searching for a sentiment score assignment to candidate lexicon entries that optimizes a well designed objective function. To design the objective function, there will be constraints defined from

different sources of information so that the optimal solution to the objective captures the intuitions behind different evidences.

Formally, suppose we are provided with a collection of m opinionated text data (or reviews for short) $D = \{d_1, d_2, ..., d_m\}$ in a given domain, k defined aspects and n candidate lexicon entries extracted from the previous step, i.e. n is the number of aspect-opinion pairs. Our goal is to compute S, a $n \times 1$ vector, where each $S_j \in [-1, 1]$ indicates the sentiment score of the aspect-opinion pair j in the given domain. For convenience, let a_j denote the aspect of j, w_j the opinion word in pair j. Basically, S_j is a concise representation of an entry in the context-dependent sentiment lexicon as defined in Section 4.2.2, i.e. $S_j = L_c(a_j, w_j)$.

Constraints for Sentiment Prior: Given an aspect-opinion pair j, if we do not have any clue about the polarity of word w_j in the special context of aspect a_j , a natural guess is w_j's sentiment score in a general-purpose sentiment lexicon (if it is in there), which should give us good prior information.

Provided with a general-purpose sentiment lexicon L, we define two $n \times 1$ vectors G and I^G: for each pair j, we set $G_j = L(w_j)$ and $I_j^G = 1$ if w_j exists in L; otherwise, $G_j = 0$ and $I_j^G = 0$. Basically, I_j^G is an indicator as whether the word w_j has prior sentiment score or not while G_j is the score if there is one available. Now, we introduce the first part of our objective function

$$\text{minimize} \left\{ \sum_{j=1}^{n} I_j^G \, |S_j - G_j| \right\} \tag{4.1}$$

This component in the objective function favors a context- dependent sentiment score assignment of S that is closest to the general-purpose sentiment lexicon, i.e. G.

Constraints for Overall Sentiment Ratings: Unlike the general-purpose sentiment lexicon that provides the prior sentiment information of words, overall sentiment ratings only represent the sentiment score at the document level. Nevertheless, it is usually assumed that the overall sentiment rating is positively correlated with the sentiment of the words in the document, which has been validated in some existing work [52, 83].

We define O as a $m \times 1$ vector, where O_i is the overall sentiment rating of the review text d_i normalized to $[-1, 1]$. Let $f(d_i, S)$ be a sentiment prediction function that outputs a sentiment score based on the review text d_i and our context-dependent sentiment lexicon S. Then we want the sentiment score calculated from our lexicon to be close to the overall sentiment rating which is observed, i.e.

$$\text{minimize} \left\{ \sum_{i=1}^{m} I_i^O \left| f(d_i, S) - O_i \right| \right\} \tag{4.2}$$

where I_i^O is again an indicator as whether O_i is defined, which offers flexibility in our framework, because not all reviews have overall sentiment rating available. Here, we choose a simple but commonly-used sentiment prediction function: averaging the sentiment scores of aspect-opinion pairs appearing in the review text based on our context-dependent sentiment lexicon. Formally, let X bs a $m \times n$ co-occurrence matrix, where each X_i is a $1 \times n$ vector representing the unigram language model of review d_i in terms of aspect-opinion pairs. In other words, X_{ij} is the number of times that the particular pair j occurs in review d_i divided by the total number of pairs in review d_i. We also take into account the "negation" rules here: If there are any negation words in the same clause, we replace the count of this occurrence from 1 to -1 when estimating X_{ij}. Then, replacing $f(d_i, S)$ with $\sum_{j=1}^{n} X_{ij} S_j$ in term (4.2), we have the following term as the second part in the objective function

$$\text{minimize} \left\{ \sum_{i=1}^{m} I_i^O \left| \sum_{j=1}^{n} X_{ij} S_j - O_i \right| \right\} \tag{4.3}$$

This term (4.3) is basically a linear regression formulation where we are looking for a solution for the unknown variables S by minimizing the distance between the observed values of the dependent variable O and the predicted values which are based on the independent variables X. matrix).

Constraints for Similar Sentiments: We can collect evidences about similar sentiments

from different sources. Consider any two aspect-opinion pairs j and k on the same aspect (i.e. $a_j = a_k$), if w_j and w_k appear as synonyms in the thesaurus, or if the pairs j and k are often concatenated with conjunctives like "and" in the corpus, we can infer that their sentiments tend to be similar.

To formalize this intuition, we define A, a $n \times n$ matrix, where $A_{jk} \in [0, 1]$ denotes our confidence about pairs j and k having similar sentiments. A simple way to construct the matrix A is to set A_{jk} to 1 if $a_j = a_k$ and either w_j, w_k are synonyms in the thesaurus or pairs j, k are conjuncted by "and" linguistic heuristic in the review text for a minimal number of times; while leaving the other elements as zeros. A more sophisticated way is to use a graded confidence score in A instead of just binary. Now we define the third part in the objective function:

$$\text{minimize} \left\{ \sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} \left| S_j - S_k \right| \right\} \tag{4.4}$$

This term (4.4) requires that whenever two paris j and k are connected in the matrix A, their sentiment scores S_j and S_k should be close.

Constraints for Opposite Sentiments: Along a similar line as the previous constraints, we define B, a $n \times n$ matrix, where $B_{jk} \in [0, 1]$ represents our confidence about pairs j and k having opposite sentiments. The value of B_{jk} where $a_j = a_k$ is based on whether w_j and w_k appear as antonyms in the thesaurus, and whether the pairs j and k are concatenated with conjunctives like "but" multiple times in the corpus.

However, the constraints of opposite sentiments are more complicated than those of similar sentiments, because we want their scores to be at the two extremes, so there is the sign of the sentiment score involved. Being opposite sentiment scores, the two scores are assumed to be in different signs (one positive and the other negative); at the same time, their absolute score values are assumed to be close.

In order to model this intuition, we separate the representation of *sign* and *absolute value* for each S_j by introducing two additional non-negative variables S_j^+ and S_j^-. We require

S_j^+ and S_j^- both to be non-negative, but at most one of them is active (i.e. positive), the other being zero. In this way, (1) which variable being active represents the sign of S_j, i.e. S_j^+ being active is equivalent to S_j being positive; S_j^- being active is equivalent to S_j being negative; and (2) the value of the active variable (S_j^+ or S_j^-) represents the absolute value of S_j.

This idea of separating the representation of S_j's sign and absolute value is implemented as follows:

$$\text{minimize} \left\{ \sum_{j=1}^{n} (S_j^+ + S_j^-) \right\} \tag{4.5}$$

subject to

$$S_j = S_j^+ - S_j^- \qquad \text{for } j = 1 \cdots n \tag{4.6}$$

$$S_j^+, S_j^- \geq 0 \qquad \text{for } j = 1 \cdots n \tag{4.7}$$

Given the equality constraints on (4.6) (4.7), term (4.5) is essentially forcing at least one of S_j^+ and S_j^- to be zero. For example, if $S_j = 0.85$ and given no other constraints, the assignment of $S_j^+ = 0.85, S_j^- = 0$ will be favored over $S_j^+ = 1, S_j^- = 0.15$, as the first assignment minimizes $(S_j^+ + S_j^-)$.

Now that we can represent the sign and absolute value of each S_j separately, we define the fourth part of the objective function as follows:

$$\text{minimize} \left\{ \sum_{j=1}^{n} \sum_{k=1}^{n} B_{jk} \left(\left| S_j^+ - S_k^- \right| + \left| S_j^- - S_k^+ \right| \right) \right\} \tag{4.8}$$

Term (4.8) favors a solution in which if two instances S_j and S_k are connected in the opposite-sentiment matrix B, their sentiment signs are different but absolute values of sentiment scores are close.

Full Objective Function

Combining all the constraints defined above, we have the following full objective function :

$$\begin{aligned}
\mathbf{\Omega} &= \frac{\lambda_{prior}}{||I^G||_1}\sum_{j=1}^{n} I_j^G \left|S_j - G_j\right| && (4.9)\\
&+ \frac{\lambda_{rating}}{||I^O||_1}\sum_{i=1}^{m} I_i^O \left|\sum_{j=1}^{n} X_{ij}S_j - O_i\right| && (4.10)\\
&+ \frac{\lambda_{sim}}{||A||_1}\sum_{j=1}^{n}\sum_{k=1}^{n} A_{jk}\left|S_j - S_k\right| && (4.11)\\
&+ \frac{\lambda_{oppo}}{||B||_1}\sum_{j=1}^{n}\sum_{k=1}^{n} B_{jk}\left(\left|S_j^+ - S_k^-\right| + \left|S_j^- - S_k^+\right|\right) && (4.12)\\
&+ \frac{\delta}{n}\sum_{j=1}^{n}(S_j^+ + S_j^-) && (4.13)
\end{aligned}$$

Now the optimization problem is

$$S = \text{argmin } \mathbf{\Omega} \tag{4.14}$$

subject to:

$$\begin{aligned}
S_j = S_j^+ - S_j^- \quad & \text{for } j = 1\cdots n\\
S_j^+, S_j^- \geq 0 \quad & \text{for } j = 1\cdots n\\
-1 \leq S_j \leq 1 \quad & \text{for } j = 1\cdots n
\end{aligned}$$

where $\lambda_{prior}, \lambda_{rating}, \lambda_{sim}, \lambda_{oppo}$ are weighting parameters which should be set to the degree that we trust each source of information, and δ can be set to a small value such as 0.01. For example, if we believe the similar-sentiment and opposite-sentiment information are of equal importance, we can set $\lambda_{sim} = \lambda_{oppo}$. The denominators in the form of $||M||_1$ represent the 1-norm of the corresponding vector or matrix M, i.e. the sum of all elements absolute values. These are constants used to normalize the weighting parameters so that their impact is comparable. Note that, it is possible to use other loss functions in the objective function

such as mean squared loss, but our specific choice can be transformed into efficient linear programming.

Transformation into Linear Programming

To solve the optimization problem efficiently, we can transform it into an equivalent linear programing problem. Basically, for each absolute-value term, we introduce one additional non-negative variable representing the non-negative absolute value. For example, we introduce $x_1, x_2, ..., x_n$ for the first part of objective function in (4.9) and replace $\sum_{j=1}^{n} I_j^G \, |S_j - G_j|$ with $\sum_{j=1}^{n} I_j^G x_j$ and two sets of additional constraints:

$$\begin{aligned} S_j - G_j &\leq x_j \qquad \text{for } j = 1 \cdots n \text{ and } I_j^G = 1 \\ -S_j + G_j &\leq x_j \qquad \text{for } j = 1 \cdots n \text{ and } I_j^G = 1 \end{aligned}$$

The additional constraints imply that $x_1, x_2, ..., x_n$ are non-negative, so we do not need to explicitly list the non-negative constraints. Similarly, we can apply similar transformation to all the other terms in the objective function and obtain a linear programming problem where the objective function, equality and inequality constraints are all linear, i.e.

$$\begin{aligned} S \quad &= \text{argmin } \mathbf{\Omega} = \text{argmin} \\ \{ \quad &\frac{\lambda_{prior}}{||I^G||_1} \sum_{j=1}^{n} I_j^G x_j + \frac{\lambda_{rating}}{||I^O||_1} \sum_{i=1}^{m} I_i^O y_i + \frac{\lambda_{sim}}{||A||_1} \sum_{j=1}^{n} \sum_{k=1}^{n} A_{jk} z_{ij} \\ + \quad &\frac{\lambda_{oppo}}{||B||_1} \sum_{j=1}^{n} \sum_{k=1}^{n} B_{jk}(u_{jk} + u_{kj}) + \frac{\delta}{n} \sum_{j=1}^{n} (S_j^+ + S_j^-) \; \} \end{aligned}$$

subject to

$$
\begin{aligned}
S_j = S_j^+ - S_j^- \qquad & \text{for } j = 1 \cdots n \\
S_j^+, S_j^- \geq 0 \qquad & \text{for } j = 1 \cdots n \\
-1 \leq S_j \leq 1 \qquad & \text{for } j = 1 \cdots n \\
S_j - G_j \leq x_j \qquad & \text{for } j = 1 \cdots n \text{ and } I_j^G = 1 \\
-S_j + G_j \leq x_j \qquad & \text{for } j = 1 \cdots n \text{ and } I_j^G = 1 \\
\sum_{j=1}^{n} X_{ij} S_j - O_i \leq y_i \qquad & \text{for } i = 1 \cdots m \text{ and } I_j^O = 1 \\
-\sum_{j=1}^{n} X_{ij} S_j + O_i \leq y_i \qquad & \text{for } i = 1 \cdots m \text{ and } I_j^O = 1 \\
S_j - S_k \leq z_{jk} \qquad & \text{for } j, k = 1 \cdots n \text{ and } A_{j,k} > 0 \\
-S_j + S_k \leq z_{jk} \qquad & \text{for } j, k = 1 \cdots n \text{ and } A_{j,k} > 0 \\
S_j^+ - S_k^- \leq u_{jk} \qquad & \text{for } j, k = 1 \cdots n \text{ and } B_{j,k} > 0 \\
-S_j^+ + S_k^- \leq u_{jk} \qquad & \text{for } j, k = 1 \cdots n \text{ and } B_{j,k} > 0
\end{aligned}
$$

An important and nice theoretic property of linear programming is that the linear constraints define the feasible region, which is a convex polyhedron; and a linear objective function is also a convex function, which implies that every local minimum is a global minimum. By transforming our optimization problem into an equivalent linear programming problem, we can utilize many known methods and toolkits to solve it efficiently. Since the construction of sentiment lexicon is an offline task, no real-time response is required. But still, all the experiments on our data sets finished within a few seconds.

4.2.5 Experiments

In this section, we present the experimental evaluation of our techniques. Our experiments employ two data sets from very different domains: one is hotel reviews from TripAdvisor

	Hotel Data	Printer Data
Domain Specific Sentiments	ROOM:private + FOOD:excelent + LOCATION:farthest - FOOD:tiny -	SOFTWARE:compatible + QUALITY:professional + ERRMSG:frequently - SUPPORT:eventually -
Aspect Dependent Sentiments	ACTIVITIES:inside - FACILITIES:inside + ROOM:huge + PRICE:huge - ACTIVITIES:cool + SERVICE:cool -	QUALITY:high + NOISE:high - INK:cheap + APPEARANCE:cheap - INK:fast + SUPPORT:fast -

Table 4.9: Sample Results of OPT

(hotel data); the other is customer feedback survey for printers (printer data). Following most previous works, we extract adjectives and adverbs as candidate opinion words, although our method is general enough to score candidate opinion words in any part-of-speech. A WordNet-based lemmatizer is employed to transform each word to its original form (e.g. "checked" to "check"). For solving the linear programming problem, we use GAMS/CPLEX, which solves our problems within a few seconds on a machine with 2.80 GHz CPU and 2GB memory. The default setting used in the proposed optimization framework (**OPT**) is $\lambda_{prior} = \lambda_{sim} = \lambda_{oppo} = \lambda_{rating}$.

As comparison, we also consider the following baselines for learning a context-dependent sentiment lexicon:

- **Random**: for each aspect-opinion pair, simply predict its sentiment by random guessing, i.e. 33.33% as positive (+1), 33.33% as negative (-1), and 33.33% as neutral (0).

- **MPQA**: for each aspect-opinion pair j, simply predict its sentiment by looking at the sentiment of the opinion word w_j in the general-purpose sentiment lexicon MPQA[8].

- **INQ**: same as the previous method, except that General Inquirer[9] is used instead of

[8] `http://www.cs.pitt.edu/mpqa/`

[9] `http://www.wjh.harvard.edu/~inquirer/`

MPQA.

- **Global**: the Global Prediction method proposed in Section 4.1. It uses only the overall ratings to generate a context-dependent sentiment lexicon with a Naive Bayes method.

Note that, we are aware of two other methods in addition to the Global method that can output aspect-dependent sentiment scores. But the idea in [12] is similar to the Global method; and the other method [83] has a strict requirement that each text should come with all k aspects, which is not realistic and does not hold in our data sets. Thus, we only include the Global method here as a representative of state-of-the-art.

Sample Results

We first present some interesting sample results in the context-dependent sentiment lexicon constructed by our optimization framework. From Table 4.9, we can see that

1. Our method picked up domain-specific new sentiment words that are not in any general-purpose sentiment lexicon. For example, "private" is positive in the hotel domain and "compatible" is positive in the printer domain. In addition, our method can detect correct sentiment even when the spelling is wrong, e.g. "excelent". That is because we consolidate different statistical evidences to infer its meaning rather just looking at the matching string in the general lexicon.

2. Even in the same domain, our method also identified different sentiments for the same word depending on the aspects. For example, in hotel reviews: "huge room" conveys positive sentiment while "huge price" is not desirable. It is negative if the activities are "inside", but it is positive if the facilities are "inside" rather than "outside". Similarly, in the printer data, "high quality" is good but "high noise" is bad. People are happy if the ink is "cheap", but they are not happy about the "cheap appearance". The word "fast" has a negative connotation for "ink" (e.g. "ink runs out fast"), but it is positive if the support service is "fast".

Evaluation of Lexicon Quality

There is no existing data set available to evaluate the quality of a constructed context-dependent sentiment lexicon, which is in the form of a sentiment score assigned to each aspect-opinion pair. In this section, we describe how we create a gold standard by performing human annotation on a data set of hotel reviews from TripAdvisor. By comparing against this gold standard, we evaluate the lexicons constructed using different methods.

Hotel Data

Data Description: We collected 4792 reviews about a well-known hotel brand from TripAdvisor. Each review has an overall rating (between 1 and 5 stars) of the hotel from the user in addition to the review text. We manually specified 7 aspects in the hotel domain, i.e., Location, Food, Room, Facilities, Service, Value and Activities. For example, the aspect or word cluster "LOCATION" contains words like: downtown, shuttle, metro, airport and etc.

Human Annotation: We randomly sample 750 reviews out of 4792 reviews to be labeled by 5 human judges, and each review is ensured to be labeled by 2 judges. For each sentence with extracted candidate aspect-opinion pairs (using the method described in Section 4.2.4), we display the original sentence to the judges followed by the tuples in the format of "aspect:attribute:opinion". The judges are asked to label each tuple with one of the following tags:

+: if positive in the context

-: if negative in the context

0: if neutral in the context

N: if do not apply

X: if attribute-aspect mapping is wrong

Below we show an instance that the judge will see.

```
"within 10 mins , we were checked in and on
our way to our room , which was fantastic."
SERVICE:check_in:fantastic
ROOM:room:fantastic
```

Note that, there may be ambiguities. In the above example, judges may have their own opinions about whether "fantastic" applies to "SERVICE" or "ROOM"or both. Considering all occurrences of aspect-opinion pairs which are labeled with +, -, or 0, the average agreement among human annotators is 78.18% which is comparable to what had been reported in existing work of sentiment analysis [67].

Gold Standard: After collecting the labels from human judges, we filter aspect-opinion pair occurrences to keep only the 3730 occurrences agreed by both judges. Then we aggregate those instances into 1127 unique pairs. To alleviate the ambiguity problem, we create our gold standard sentiment lexicon by using only the 705 aspect-opinion pairs labeled +1 or -1, which tend to represent high confidence and consistency of the labels. This gold standard lexicon is domain specific and aspect-dependent as well; it contains high-quality entries agreed by human annotators. But the coverage is relatively small because we only include the high-confident ones in the gold standard in order to be accurate.

- Evaluation Measures Since the gold standard sentiment lexicon contains only binary labels (either +1 or -1), we first transform our output sentiment lexicon into the same format by only considering the sign of the predicted sentiment value, so that the assigned scores are either +1 or -1. After that, the output sentiment lexicon can be evaluated by:

$$\begin{aligned} \textbf{precision} &= \frac{N_{agree}}{N_{lexicon}} \\ \textbf{recall} &= \frac{N_{agree}}{N_{gold}} \\ \textbf{F-measure} &= \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \end{aligned}$$

Method	Precision	Recall	F-Measure
Random	0.4932	0.2784	0.3559
MPQA	**0.9631**	0.3702	0.5348
INQ	0.8757	0.4397	0.5855
Global	0.7073	0.5929	0.6451
OPT	0.8125	**0.6823**	**0.7417**

Table 4.10: Lexicon Quality Evaluation on Hotel Data

where N_{gold} is the number of aspect-opinion pairs in the gold standard lexicon, $N_{lexicon}$ is the number of aspect-opinion pairs in the automatically constructed sentiment lexicon (i.e. 705), N_{agree} is the number of pairs that are consistently labeled (either both +1 or both -1) in the gold standard and constructed lexicons.

- Results

  Note that the human annotation is for evaluation purpose only, and the automatic algorithms do not use any labels. So we run the algorithms on the whole set of 4792 reviews instead of the subset of 750 reviews labeled by human judges. After generating candidate lexicon entries, we extract 4627 unique aspect-opinion pairs with at least two occurrences, and score them with different algorithms. However, as there are only 705 pairs in the gold standard, there is some bias in the evaluation by precision. This is because of the fact that there can be some aspect-opinion pairs correctly output by the algorithms but they do not appear in the subset of 750 reviews so human annotators did not label them. As a result, the precision should be taken with a grain of salt here. Take an extreme example: a naive method outputting only one correct pair (e.g. "LOCATION:excellent:+1") will have 100% precision but extremely low recall; but it is not useful in practice. Thus, F-measure should be a more reliably measure in order to evaluate the usefulness of a sentiment lexicon, because it captures the balance between precision and recall.

  The results of different methods on hotel data are shown in Table 4.10 where the best performance under each measure is highlighted in bold font. We can see that when

directly evaluating the lexicon quality,

1. Dictionary-based baselines (i.e. MPQA and INQ) which totally ignore the context, provide best precision performance, at the price of low recall. The recall of MPQA and INQ is significantly lower than other methods that take context into consideration (Global and OPT). This suggests that there are a lot of domain specific and aspect dependent words that carry sentiments but are totally ignored by dictionary-based baselines.
2. In comparison, the Global method, gives a better balance of precision and recall and thus better F-measure. This method is able to pick up domain specific and context dependent sentiments by exploiting the association among aspects, words and document-level overall rating.
3. Our method OPT further improves the Global method in *both precision and recall* significantly (and thus F-measure too, by almost 15%). This is because that in addition to the overall rating OPT also incorporates the prior sentiments from dictionaries, the similar/opposite sentiment information and linguistic heuristics, which help the sentiment prediction especially when the signal from the overall ratings is not present or not strong enough to tell the sentiments of some words.

Evaluation of Aspect-Level Sentiment Classification Using the Lexicon

The value of a sentiment lexicon mostly lies in its use in applications. Thus, in addition to the evaluation of the lexicon quality, we also conduct experiments to evaluate aspect-level sentiment classification performance of using different lexicons. The task is to produce a sentiment score for a given aspect in a piece of text, e.g. whether a particular hotel review is talking positively or negatively about the LOCATION aspect.

- Hotel Data

From the manually annotated hotel data described in Section 4.2.5, we use the sentiments at each review-aspect level as the gold standard. Again, in order to ensure the high confidence of the gold standard, we only consider those aspect-opinion pairs that have labels agreed by two judges. After that, the gold standard sentiment at each review-aspect level is the averaged sentiment labels of the corresponding aspect-opinion pairs in the review, which is a real value between -1 and $+1$.

- Printer Data

  Data Description: For the second data set, we obtain 3511 customer feedback surveys about a printer brand. Each survey comes with an overall satisfaction rating (between 1 and 5) and a small piece of text of detailed comments (usually just one or two sentences).

  Human Annotation: The company manufacturing the printers hired people to manually label the feedback text so as to get deeper understanding about what people are happy about their printers and what they are upset about. The human judges are provided with an aspect description file, in which a set of aspect tags are defined by a short description. For example

```
[TRIES]: The number of unsuccessful tries
         before install success.
[INK]: Ink and print head related issues
       (Including Install and Removal).
```

  During the labeling process, the judges read each survey, tag it with the matching aspect tags, and assign a sentiment score among $\{-3, -2, -1, +1, +2, +3\}$ for each aspect tag. For instance, the review text of "Easy to set up. digital monitoring is great for ink needs. " is tagged as "[+3, TRIES] " and "[+3, INK]", because it is

Statistics	Hotel Data	Printer Data
# of reviews	750	3511
# of possible aspects	7	25
AVG # of aspects per review	2.86	1.32
AVG # words per review	270	24

Table 4.11: Data Set Statistics for Sentiment Classification Task

talking very positively about both the "TRIES" and the "INK" aspects. Then we use the top 25 most frequently tagged aspects in our experiments. Unfortunately, we do not know further details such as how many human judges are involved and what is their agreement, so we cannot report them here.

Both the hotel data and the printer data are manually labeled with different sentiment scores for each document-aspect combination. This enables us to evaluate the aspect-level sentiment classification performance of using different sentiment lexicons, which represents a real application need. Actually the classification results are essentially what the printer company is interested in. If we can do accurate classification automatically, we can save companies effort to hire people to label the aspect-level sentiment. Some statistics about the two data sets are summarized in Table 4.11.

- Evaluation Scheme and Measures For the task of sentiment classification at the document-aspect level, we need to first use a sentiment lexicon to predict the sentiment score for each document-aspect combination. Since we only use an unlabeled corpus, we will continue using unsupervised method for the prediction. In particular, we adopt the following simple but reasonable baseline approach: for each document-aspect combination (d_i, a_j),we identify all the aspect-opinion pairs on the aspect a_j occurring in document d_i, look up the sentiment score of each pair in the context-dependent sentiment lexicon, and then take the average of sentiment scores as the predicted score for this combination (d_i, a_j).

  Now if we only consider the binary sign of the sentiment scores, we can also use

Method	Prec	Recall	F-Measure	MSE
Hotel Data				
Random	0.4368	0.3689	0.3999	0.567
MPQA	**0.8128**	0.5289	0.6408	0.47
INQ	0.78	0.6294	0.6966	0.4561
Global	0.6975	0.773	0.7333	0.4426
OPT	0.7283	**0.7756**	**0.7512**	**0.416**
Printer Data				
Random	0.4844	0.2629	0.3408	0.7142
MPQA	0.7579	0.1597	0.2639	0.574
INQ	0.7879	0.3502	0.4849	0.5365
Global	0.7645	**0.5448**	0.6362	0.5091
OPT	**0.8222**	0.5276	**0.6428**	**0.468**

Table 4.12: Sentiment Classification Performance

precision, recall, and F-measure for evaluation. But as the gold standard scores are real values (all normalized to $[-1, 1]$ by min-max normalization) rather than being binary, we also include **Mean Squared Error (MSE)** as an additional measure, which measures the distance between the predicted sentiment and the gold standard sentiment. MSE is more an accurate measure in the sense that it captures the notion that classifying a positive class into a negative class is worse than classifying it into a neutral one. Lower MSE means better classification accuracy.

- Results

  We summarize the results on both data sets in Table 4.12 and highlighted in bold font the best performance under each measure.

  In the aspect-level sentiment classification task, which is a real application of the constructed context-dependent sentiment lexicon,

  1. dictionary-based baselines (MPQA and INQ) do not necessarily gives best precision. Moreover, they suffer more at recall on the printer data. (Especially, recall of MPQA is even lower than the random baseline.)
  2. The Global method still performs well on both precision and recall.

3. Our OPT method provides the best balance between precision and recall; it achieves the best F-measure performance on both data sets.

4. Furthermore, when we zoom into the performance evaluated at finer granularity, i.e. as measured by MSE, the performance gain of OPT is even more significant. It has reduced the best MSE in the baselines from 0.4426 to 0.416, from 0.5091 to 0.468 on the two data sets respectively, both improvements are statistically significant with p-value less than 10^{-6} in a paired t-test.

All these observations suggest that a lexicon with higher precision (as shown by dictionary-based baselines in Table 4.10 where we directly evaluate the lexicon quality) does not necessarily lead to better aspect-level classification performance. The low recall of the dictionary-based baselines would result in many misses of domain-specific and aspect-dependent polarity words, thus lead to less accurate classification of aspect-level sentiment. Thus, it is important to achieve a good balance between precision and recall. In particular, if one is mainly interested in aspect-level classification, which is one of the most important applications of sentiment lexicons, OPT is by far the best method. Such performance advantage demonstrates the effectiveness of combining multiple useful signals in our optimization framework.

Analysis of Parameter Tuning

We have already shown that OPT in the default parameter setting outperforms all baselines on both lexicon quality evaluation and sentiment classification evaluation. Now we further look into the four parameters $\lambda_{prior}, \lambda_{sim}, \lambda_{oppo}, \lambda_{rating}$ that basically weight the importance of the four components in the objective function. Our framework is very general, and if we set one parameter to zero it is equivalent to not using the signal as defined in the corresponding term. For the purpose of examining the importance of different signals, we conduct some analysis experiments where one term is dropped out in each experiment.

	λ_{prior}	λ_{rating}	λ_{sim}	λ_{oppo}	**F-Measure**
Default	1	1	1	1	0.7417
Drop	0	1	1	1	0.6549
one	1	0	1	1	0.6453
term	1	1	0	1	0.7309
	1	1	1	0	0.7408
Weighting	2	2	1	1	0.7431
important	3	3	1	1	**0.7544**
terms	6	6	1	1	0.7510
	8	8	1	1	0.7506

Table 4.13: OPT Parameter Tuning: Lexicon Quality on Hotel Data

Lexicon Quality: The middle rows in Table 4.13 show the lexicon quality evaluation results of "dropping one term" tested on the hotel data. Due to the space limit, we only display the F-measure here. It can be seen that (1) dropping any term in the objective function decreases the lexicon quality, indicating that all the constraints are useful. (2) when setting λ_{prior} or λ_{rating} to zero, the performance decreases dramatically (F-measure from 0.7417 to around 0.65), which suggests that these two terms contain more important information. Then we tried to place more weights on the two important terms. As shown in the bottom four rows, performance can be further increased, where the best one is highlighted in bold font.

Classification Performance: In Table 4.14, we also show results of parameter tuning on the sentiment classification task. Similar trend is observed too, i.e. classification performance is improved if we put more weights on the important signals. One thing to note is that the importance of signals is different in the two data sets: both the prior sentiments and the overall ratings are important in the hotel data while the overall ratings serve as the most important signal in printer data.

This series of experiments demonstrate that our optimization framework is general enough to accommodate different weights placed on different kinds of signals for constructing a context-dependent sentiment lexicon, which can lead to even better performance than the default setting. This is especially useful when we have some reliable prior belief of the

importance of signals; then we can put more weights on more important signals. Nevertheless, there is still the challenge of automatically setting the optimal parameters for different domains and/or different data sets, which we intend to study as future work.

4.2.6 Conclusion and Future Work

In this chapter we studied the problem of automatically constructing a context-dependent sentiment lexicon from an unlabeled opinionated text collection. We studied and summarized several kinds of useful signals, formulated an optimization problem to combine all the signals, and provided a mathematical transformation into linear programming. We have demonstrated that our method can learn new domain specific sentiment words and aspect-dependent sentiment. Further quantitative evaluation against baselines and a state-of-the-art method shows that (1) for a given domain our framework can greatly improve the coverage of a general sentiment lexicon; (2) constructed aspect-level sentiment lexicons are in good quality, achieving a good balance of precision and recall; (3) aspect level sentiment classification performance can be significantly improved with the automatically constructed context-dependent sentiment lexicon; and (4) parameter tuning gives more performance advantage.

The framework we proposed is quite general and applicable for opinionated text collection in any domain. It is capable of incorporating different sources of available information for the automatic construction of a context-aware sentiment lexicon. As future work, we can exploit other kinds of useful signals such as "pros" and "cons" sections in the reviews and aspect-level ratings. We also plan to evaluate the effectiveness of our context-aware sentiment lexicon in other sentiment related applications, such as opinion retrieval and opinion summarization. Another interesting future work is to study how to tune the weighting parameters automatically for optimal performance.

					Hotel	Data					Printer	Data
	λ_{prior}	λ_{rating}	λ_{sim}	λ_{oppo}	**F-Measure**	**MSE**	λ_{prior}	λ_{rating}	λ_{sim}	λ_{oppo}	**F-Measure**	**MSE**
Default	1	1	1	1	0.7512	0.416	1	1	1	1	0.643	0.468
Drop	0	1	1	1	0.7396	0.4436	0	1	1	1	0.656	0.467
one	1	0	1	1	0.6629	0.4749	1	0	1	1	0.453	0.673
term	1	1	0	1	0.7733	0.4057	1	1	0	1	0.657	0.446
	1	1	1	0	0.7508	0.4132	1	1	1	0	0.649	0.468
Weighting	2	2	1	1	0.7632	0.4096	1	2	1	1	0.662	0.459
important	3	3	1	1	0.7737	0.4054	1	3	1	1	0.668	0.456
terms	6	6	1	1	0.7781	0.4015	1	6	1	1	0.671	0.451
	8	8	1	1	**0.7794**	**0.4008**	1	8	1	1	**0.672**	**0.449**

Table 4.14: OPT Parameter Tuning: Sentiment Classification Performance on Both Data Sets

Chapter 5

User Level Sentiment Analysis

In the previous Chapter, we introduced two new methods for aspect level sentiment analysis. These methods are very general without requiring human supervision. Instead, they utilize "free" information when available, such as the overall sentiment ratings, general-purpose sentiment lexicon, synonym/antonym dictionary and linguistic heuristics. However, these methods may not work as well in some difficult topic domains, such as political discussions, where the difficulty comes from the sarcasm and complicated background knowledge involved. Moreover, so far each piece of opinion text is treated as equivalent while apparently the opinion holder is also very important. For example, whether a Computer Science PhD student is pro-choice in the abortion issue cannot be counted the same as whether President Obama is.

To this end, in this chapter we study user level sentiment analysis. In particular, we choose the domain of political forum discussions. This domain is known for its difficulty to get accurate sentiment analysis results, no to mention unsupervised sentiment analysis, which is our focus here. This raises interesting challenges, which we will address by combining textual content analysis (e.g. post content) and social network analysis (e.g. who says what and who talks to whom).

5.1 Overview

Online forums, which date back as far as 1994 [4], is one of the early applications managing and promoting user generated content [3, 2]. Although being simple in its design – users

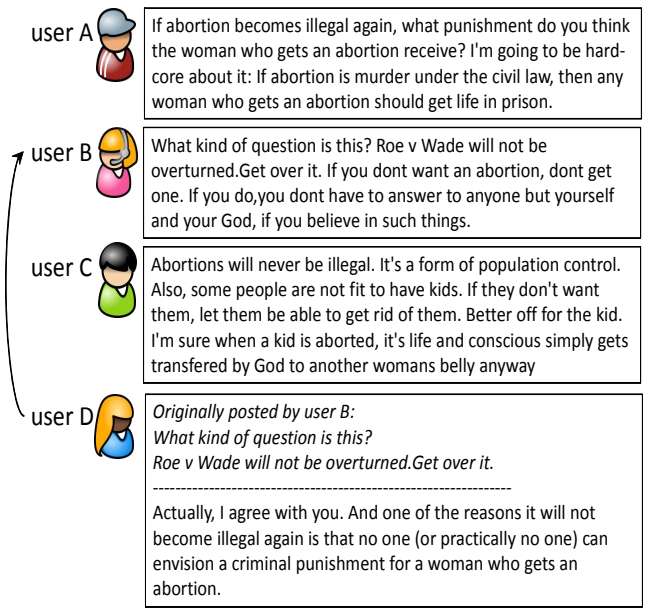


Figure 5.1: Illustration of a Forum Thread on "Abortion"

carry out discussion in the form of message threads (An example of a forum thread on "abortion" is illustrated in Figure 5.1), forums remain prevalent and popular even during the recent rise of many sophisticated Web 2.0 applications. For instance, Japanese people post most frequently with over two million per day on their largest forum, 2channel (`http://www.2ch.net`). China also has millions of posts on forums such as Tianya Club (`http://www.tianya.cn`). As users actively express their opinions and exchange their discussions on all kinds of topics/issues, e.g. technology, games, sports, music, fashion, religion, and politics, forums are becoming a great source for opinion mining. However, the simple design of forums combined with rapidly accumulated data make it challenging to make sense out of the forum discussions.

As time goes by, some regular forum users may develop a sense of "virtual community" [44] which can be considered as a type of hidden social network. When formed, this sense of community is very helpful in future forum activities. For example, if we know some users are unreasonably biased toward something (e.g. President Barack Obama or the Apple company) from history discussions, we may think twice about their opinion about similar

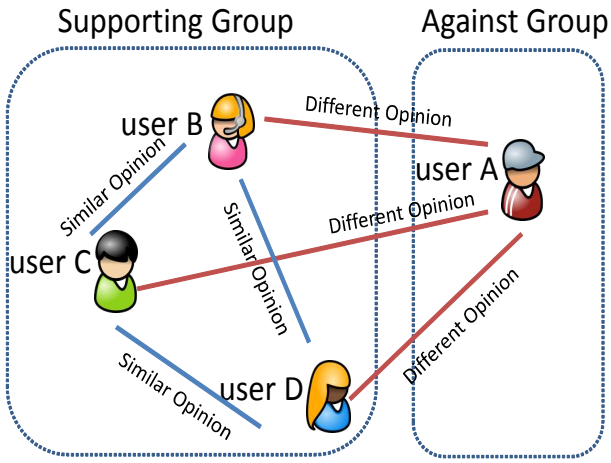


Figure 5.2: Example Opposing Opinion Network for the Thread on "Abortion"

topics; or, we may even cite their historical unreasonableness if we want to argue against them. However, such sense of virtual community can only be formed by accumulated effort of reading and participating in lots of forum discussions. This is very difficult to achieve by ordinary readers and even occasional forum users, who can only capture a local view of the posts without any context or background information.

To this end, we propose a new problem of automatically discovering *opinion networks* from forum discussions, which is a latent social network with links based on user opinions on different topics. Such discovered opinion networks not only serve as a concise summary of the forum but also provide a sense of virtual community for any online user. In general, such an opinion network is a graph with users as nodes and edges indicating their relations in terms of their opinions. In this chapter, we study a special case of *opposing opinion networks* where there are only two sets of opposing users for each topic: a supporting group and an against group. (See an example illustration in Figure 5.2) The identified opposing opinion networks can enable a number of interesting applications, for example (1) detecting threads of heated debate; (2) detecting users of similar minds who often agree with each other across different topics; (3) detecting "enemy" users who often argue with each other across different topics; (4) detecting similar topics which involve similar groups of user interactions; (5) contrastive summarization of topical discussions.

Discover the opposing opinion networks is related to some existing work on opinion mining, but most proposed approaches are supervised [23, 76]. Instead, we are interested in an unsupervised approach so that we will be able to easily apply our method to forums in any domain without requiring labeled training data. Treating each forum user as a big document containing all her posts would lose the rich information around each post. Instead, our basic idea is to first infer the opinion in each post and then get the user opinion as the aggregation of the opinions in all her posts. This kind of approach is able to consolidate rich local information (about posts, reply-to relation, etc) for better prediction of global user opinion. In inferring opinions in forums, most previous work either use the link information without using any of the content information [7] or use the context content without much consideration of users social interaction [23]. However, we will show the power of combining content analysis and social network analysis together.

In particular, we propose to exploit the unique characteristics of forum data and analyze signals from both textual content (e.g. post content) and social interactions (e.g. who talks to whom). More specifically, we propose two assumptions from social network analysis: (1) user consistency and (2) user relation consistency; two kinds of analysis from content part: (1) topic model analysis of aspect mentions in posts and (2) bootstrapping-based classification of agree/disagree relations between posts. Finally, to consolidate all the signals together, we design an optimization formulation and transform it into linear programming so as to solve it efficiently.

Discovering opinion networks is a new problem and there is no existing data set that can be used for evaluation. To solve this problem, we created a new data set with the help from both our colleagues and crowd sourcing annotators. Experiments show that the proposed optimization method outperforms several baselines and existing approaches, demonstrating the power of combining both text analysis and social network analysis in analyzing and generating opinion networks. We also demonstrate two interesting applications of the discovered opinion networks in finding semantically similar topics and recommending similar-minded

users, respectively.

5.2 Problem Formulation

A forum F is a set of threads of discussions, i.e. $F = \{TH_1, TH_2, \cdots\}$. And each thread $TH = (P, R) \in F$ is composed of a sequence of posts $P = \{d_1, d_2, \cdots, d_n\}$ and the (partial) reply-to relation between posts $R \subset P \times P$, where $R_{ij} = 1$ if d_i replies to d_j.

In an online forum, opinion network is the latent social network where the connections are based on user opinions on different issues. More formally,

Definition (Opinion Network): An opinion network is a multi-graph (U, E) among forum users $U = \{u_1, u_2, \cdots, u_m\}$ and each edge $(u_i, u_j, t, a_{ij}^t) \in E$ carries an agreement weight $a_{ij}^t \in [-1, 1]$ conditioned on an issue t. The higher the agreement weight, the more strongly the two users share similar opinions. For simplification, we can consider each thread as discussing an issue.

An opposing opinion network is a special case of opinion network where we are only interested in the two opposing user groups for an issue.

Definition (Opposing Opinion Network): An opposing opinion network is an opinion network (U, E), and $U = U^+ \cup U^- \cup U^0$, where we are only interested in the supporting group U^+ and the against group U^- without caring of the other users in U^0; $E = \{(u_i, u_j, t, a_{ij}^t)\}$ where the edge weights are derived as follows.

$$
\begin{aligned}
a_{ij}^t &= -1 \quad if\ u_i \in U^+\ and\ u_j \in U^- \\
a_{ij}^t &= -1 \quad if\ u_i \in U^-\ and\ u_j \in U^+ \\
a_{ij}^t &= 1 \quad if\ u_i \in U^+\ and\ u_j \in U^+ \\
a_{ij}^t &= 1 \quad if\ u_i \in U^-\ and\ u_j \in U^-
\end{aligned}
$$

In this chapter, our goal is to automatically identify opposing opinion network from forum discussions.

5.3 Method Overview

It is not trivial to identify each user's opinion for any given issue directly, especially in an unsupervised way.

Baseline 1: UserClustering A natural baseline is to represent each user by concatenating all her posts and then apply clustering algorithms to separate them into two groups. However, this method (i.e., UserClustering) can only analyze the discussion content independently, failing to consider the rich social interaction context into consideration.

Baseline 2: SentiWordNet Another option is to apply sentiment tagging (using a lexicon like SentiWordNet, or opinion classifier) to the post content produced by the user and then the positive/negative results will be mapped to the supporting/against groups. However, this method (i.e., SentiWordNet) will not perform well because: (1) in online forum discussions, users do not always use positive/negative sentiment words or phrases to express their opinions. They also use a lot of domain dependent, sarcastic expressions which cannot be captured in predefined sentiment lexicons. (2) forum users do not always directly express their opinions toward an issue. More often, they interact and argue with each other and express their opinions toward other users.

Our Approach: In this chapter, we design and introduce a new approach that can handle these challenges by exploiting the complimentary information from both discussion content and social interactions within the forum data. At a high level, we propose to first identify the opinion in each post $v(d_i) \in [-1, 1]$ (or v_i for short) in a given thread, for which we incorporate multiple information: the author, the textual content, and other users who interact with the author. After that, each user's opinion is the aggregated opinion from all

her posts. We denote M_j as the authorship vector for user u_j where $M_j(i) = 1$ is post d_i was written by u_j. Then $o_j = \frac{\mathbf{v}^T M_j}{\sum_{i=1}^{n} M_j(i)}$ is the u_j's aggregated opinion in this thread. Then, given threshold $\alpha \in [0, 1]$, we have a supporting group and an against group:

$$\begin{aligned} U^+ &= \{u_j | \frac{\mathbf{v}^T M_j}{\sum_{i=1}^{n} M_j(i)} > \alpha\} \\ U^- &= \{u_j | \frac{\mathbf{v}^T M_j}{\sum_{i=1}^{n} M_j(i)} < -\alpha\} \end{aligned}$$

Or, we can simply rank the users by their opinion scores and leave the freedom to the applications as where to cutoff a certain number of top ranked users as the supporting group and bottom ranked users as the against group. For example, showing the top 10% of users that are most strongly supporting/against an issue.

Now, the problem is reduced to identifying opinions in each post, i.e., assigning an opinion score in $[-1, 1]$ to each post d_i as the degree of support (positive) or against (negative) the issue in the given thread. In the next section, we will discuss different signals in detail that help infer the opinion in each post and then introduce an optimization formulation to consolidate the signals.

5.4 Identify Opinions in Posts

As discussed before, there are lots of rich information around a post, e.g. the textual content, the author, the reply-to structure. In this section, we will analyze these rich information for better inferring the opinion in each post.

5.4.1 Analysis of Social Interactions

We have two assumptions from the point view of social network analysis:

1. *User Consistency*: in one thread, different posts from the same user tend to express consistent opinion. More specifically, suppose we know one post from u_i show strong

support for a given issue; then we assume that all the other posts written by u_i in this thread follow this support opinion.

2. *User Relation Consistency*: in one thread, two users usually stay agreed/disagreed consistently. More specifically, suppose we know one post from u_i replying to a post from u_j expresses agreement; then we assume that all the other reply-to posts between u_i and u_j also follow the agreement attitude.

Both assumptions provide useful constraints or evidence when we infer the opinion in each post.

5.4.2 Analysis of Textual Content

There are two possible parts in the textual content of a post: a mandatory statement part and an optional quotation part if it replies to a previous post. We show that there are useful signals in both parts for inferring opinions.

1. *Statement*: This is a mandatory part of the post, which states what the author wants to say. Classical opinion techniques can be applied to this field to extract the user's opinion towards a particular topic. It is also found that users with different sentiments/positions would focus on different aspects of the topic, which is called "framing" [26]. For example, on the abortion issue, pro-choice people would emphasize women's rights and freedom while pro-life people would focus on the crude process of abortion. Apparently, these two opposing groups of users tend to share similar mentions of aspects within the group and different mentions between groups.

2. *Quotation*: This is an optional part of a post when the author quotes some statements from some previous post before expressing their own opinions. This quotation part is usually visualized using different font or color in online forums, so that the readers have better sense of the context of discussion. In this kind of interaction format, the

authors usually directly express their attitude toward the quoted post/user in the first sentence of the reply. It would be a very strong indicator of users' relation if we can automatically classify each sentence as showing users agreement/disagreement, e.g. "totally!" is agreement or "it doesn't make sense to me." is disagreement.

5.4.3 Measuring Agreement/Disagreement Between Posts

We have analyzed signals from both social interactions and textual content. They are all good indicators of the relations among posts. Here, we will discuss how to concretely measure this kind of relation as agreement or disagreement.

1. *Using "user consistency":* Following user consistency assumption, we can set a matrix $A \subseteq P \times P$ from a given thread, indicating agreement relation among posts written by the same author, where $A_{i,j} = 1$ iff d_i and d_j are posts from the same user, i.e.,

$$A_{i,j} = 1,\ if\ user(d_i) = user(d_j)$$

2. *Using "framing":* We employ topic modeling approach [32] to extract the hidden aspects of discussion, so that we get a number (e.g. five) of aspect models $p(w|\theta)$ for each thread and an aspect distribution $p(\theta|d)$ for each post in this thread. Then, given any two posts from the same thread, if the two corresponding aspect distributions have positive correlation, their opinions tend to agree; otherwise, their opinions tend to disagree, because in "framing" people with different opinions would focus on different aspects of that topic. Denoting $corr(d_i, d_j) = correlation(p(\theta|d_i), p(\theta|d_j))$ as the Pearson correlation coefficients, we can have another measure of post-post relations $P \times P$ as agreement $T^{agr} \subseteq P \times P$ and disagreement $T^{dis} \subseteq P \times P$, using the following

equations:

$$T_{i,j}^{agr} = \quad corr(d_i, d_j), \quad if\ corr(d_i, d_j) > 0.5$$
$$T_{i,j}^{dis} = \ -corr(d_i, d_j), \quad if\ corr(d_i, d_j) < -0.5$$

3. *Using "user relation consistency" and "reply-to sentence":* With no labeled training data, classifying a reply-to text as agreement or disagreement is not a trivial task. Since users use various ways to express their attitude toward the quoted post, it is impossible to enumerate all the patterns beforehand. To solve this challenge, we design a bootstrapping method to classify the reply-to text by taking the advantage of whole forum analysis of user relation consistency.

   We first extract all the "reply-to sentences", i.e., the first sentence of the reply text, from the whole forum of more than 1 million posts, and then label these sentences with only a few agreement/disagreement patterns (six in total), such as "I agree" and "I disagree". After that, our idea is to bootstrap other patterns with the help of "user relation consistency": suppose we observe one post from u_i replies to a post from u_j and matches the initial "agree" pattern; then we assume that all other reply-to sentence between u_i and u_j also follow the "agree" attitude. In this way, we can extract all the "agree" sentences P^{agr} and "disagree" sentences P^{dis} from the whole forum. Essentially, we rely on the users themselves to get the different ways of expressing agreement or disagreement.

   Now, given a new reply-to sentence t_{ij} (indicating that post d_i replies to post d_j), we can just compare the $Sim(t_{ij}, P^{agr})$ versus $Sim(t_{ij}, P^{dis})$ so as to infer the attitude of t_{ij}. Here, $Sim(x, y)$ outputs a value between 0 and 1 which is the max cosine similarity of a text and a set of text. We can now mark some of the reply-to relations in R as

agree $R^{agr} \subseteq R$ and some as disagree $R^{dis} \subseteq R$, using the following equations:

$$R^{agr}_{i,j} = Sim(t_{ij}, P^{agr}), \quad if \ \frac{Sim(t_{ij}, P^{agr})}{Sim(t_{ij}, P^{dis})} \geq 2$$
$$R^{dis}_{i,j} = Sim(t_{ij}, P^{dis}), \quad if \ \frac{Sim(t_{ij}, P^{agr})}{Sim(t_{ij}, P^{dis})} \leq \frac{1}{2}$$

5.4.4 Optimization Formulation

We have introduced and analyzed different signals that can indicate the opinions in forum posts, but it is still not clear how we can combine multiple signals. One way to combine these signals is to use the agree/disagree information as distance measures between posts and then apply clustering-like methods, e.g. MaxCut as in [62]. However, there are two disadvantages: first, the clustering or partition results cannot tell which group is supporting and which is against; second, a hard partition of the users cannot tell users with strong support/against opinions from those with balanced view.

To this end, we propose a flexible optimization formulation that tries to find opinion assignment to each post $v(d_i) \in [-1, 1]$ (or v_i for short) so that they capture the different signals introduced before.

Capturing Agreement

We have constructed matrices A, R^{agr} and T^{agr} to encode the signals indicating agreement relation between posts. So, we want to minimize the opinion score difference of two posts if the corresponding entry is active in one of the matrices, i.e.,

$$minimize \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R^{agr}_{i,j} + T^{agr}_{i,j} + A_{i,j})|v_i - v_j|$$

Basically, we are giving a linear penalty if the two opinion scores differ a lot.

Capturing Disagreement

We have constructed matrices R^{dis} and T^{dis} to encode the signals indicating disagreement relation between posts. To capture such disagreement, we first separate the representation of "sign" and "absolute value" in each opinion score v_i; we can do this by introducing two non-negative variables v_i^+, v_i^- and a constraint $v_i = v_i^+ - v_i^-$. In order to ensure that no more than one of v_i^+, v_i^- is positive (the other being zero), we also need to minimize $(v_i^+ + v_i^-)$.

In this way: v_i being positive is equivalent to v_i^+ being positive and v_i^- being zero; v_i being negative is equivalent to v_i^+ being zero and v_i^- being positive; v_i being zero is equivalent to v_i^+ and v_i^- both being zero.

If there is an entry (i, j) active in one of the matrices R^{dis} or T^{dis}, we want to make the two corresponding opinion scores v_i and v_j to have opposite sign and similar absolute value. Now, we can capture that by the following terms and constraints

$$\begin{aligned} minimize \qquad & \left\{ \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{i,j}^{dis} + T_{i,j}^{dis}) \left(|v_i^+ - v_j^-| + |v_i^- - v_j^+| \right) \right. \\ + \quad & \left. \mu \sum_{i=1}^{n} (v_i^+ + v_i^-) \right\} \end{aligned}$$

subject to

$$\begin{aligned} & \forall i \in \{1, \cdots, n\}, \qquad v_i = v_i^+ - v_i^- \\ & \forall i \in \{1, \cdots, n\}, \qquad v_i^+,\ v_i^- \geq 0 \end{aligned}$$

Capturing Sentiment Priors

Agreement and disagreement relations between posts are very useful, but we still need some hint about absolute opinion in each post in order to tell which group is supporting and which is against. We turn to the sentiment tagging baseline which returns $\mathbf{s}$, sentiment scores for

the posts using SentiWordNet.

As we discussed before, forum users do not always use positive/negative words, s o many of the sentiment scores in $\mathbf{s}$ are close to zero. But those entries in $\mathbf{s}$ with scores close to -1 or 1 captures the small subset of posts containing many positive/negative words. Then, using the following term, we ensure that our opinion assignment does not deviate too much from the sentiment tagging especially when the sentiment score is high/confident.

$$minimize \sum_{i=1}^{n} |v_i - s_i|$$

Full Objective Function

Putting everything together, we have the following objective function, where λs are the weights to trade off different components.

$$\begin{aligned}
\mathbf{v} &= argmin_{\mathbf{v}} \Omega(\mathbf{v}) \\
&= argmin_{\mathbf{v}} \Bigg\{ \lambda_{senti} \sum_{i=1}^{n} |v_i - s_i| \\
+ \quad & \lambda_{agr} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{i,j}^{agr} + T_{i,j}^{agr} + A_{i,j}) |v_i - v_j| \\
+ \quad & \lambda_{dis} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{i,j}^{dis} + T_{i,j}^{dis}) (|v_i^+ - v_j^-| + |v_i^- - v_j^+|) \\
+ \quad & \mu \sum_{i=1}^{n} (v_i^+ + v_i^-) \Bigg\}
\end{aligned} \tag{5.1}$$

subject to

$$\begin{aligned}
&\forall i \in \{1, \cdots, n\}, \quad && -1 \leq v_i \leq 1 \\
&\forall i \in \{1, \cdots, n\}, \quad && v_i = v_i^+ - v_i^- \\
&\forall i \in \{1, \cdots, n\}, \quad && v_i^+,\ v_i^- \geq 0
\end{aligned}$$

where A, R^{agr}, R^{dis}, T^{agr} and T^{dis} matrices are obtained as described in previous sections, while $\mathbf{v}$, $\mathbf{v}^+$ and $\mathbf{v}^-$ are the variables.

Transformation into Linear Programming

To solve the optimization problem efficiently, we can transform it into an equivalent linear programing problem. Basically, for each term with absolute-value, we introduce one additional non-negative variable representing the non-negative absolute value. For example, we introduce $y_1, y_2, ..., y_n$ for the first term of full objective function and replace $\sum_{j=1}^{n} |v_i - s_i|$ with $\sum_{j=1}^{n} y_i$ and two sets of additional constraints:

$$\forall i \in \{1, \cdots, n\}, \qquad (v_i^+ - v_i^-) - s_i \leq y_i$$
$$\forall i \in \{1, \cdots, n\}, \qquad -(v_i^+ - v_i^-) + s_i \leq y_i$$

The additional constraints imply that $y_1, y_2, ..., y_n$ are non-negative, so we do not need to explicitly list the non-negative constraints. We apply similar transformation to all the other terms in the objective function and obtain a linear programming problem where the objective function, equality and inequality constraints are all linear, i.e.

$$\begin{aligned}
\mathbf{v} \quad &= argmin_{\mathbf{v}} \Bigg\{ \lambda_{senti} \sum_{i=1}^{n} y_i \\
+ \quad & \lambda_{agr} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{i,j}^{agr} + T_{i,j}^{agr} + A_{i,j}) z_{ij} \\
+ \quad & \lambda_{dis} \sum_{i=1}^{n} \sum_{j=i+1}^{n} (R_{i,j}^{dis} + T_{i,j}^{dis})(a_{ij} + b_{ij}) \\
+ \quad & \mu \sum_{i=1}^{n} (v_i^+ + v_i^-) \Bigg\}
\end{aligned} \tag{5.2}$$

subject to

$$
\begin{aligned}
\forall i \in \{1, \cdots, n\}, \quad & -1 \leq v_i \leq 1 \\
\forall i \in \{1, \cdots, n\}, \quad & v_i = v_i^+ - v_i^- \\
\forall i \in \{1, \cdots, n\}, \quad & v_i^+,\ v_i^- \geq 0 \\
\forall i \in \{1, \cdots, n\}, \quad & v_i - s_i \leq y_i \\
\forall i \in \{1, \cdots, n\}, \quad & -v_i + s_i \leq y_i \\
\forall i, j \in \{1, \cdots, n\}, \quad & v_i - v_j \leq z_{ij} \\
\forall i, j \in \{1, \cdots, n\}, \quad & -v_i + v_j \leq z_{ij} \\
\forall i, j \in \{1, \cdots, n\}, \quad & v_i^+ - v_j^- \leq a_{ij} \\
\forall i, j \in \{1, \cdots, n\}, \quad & -v_i^+ + v_j^- \leq a_{ij} \\
\forall i, j \in \{1, \cdots, n\}, \quad & v_i^- - v_j^+ \leq b_{ij} \\
\forall i, j \in \{1, \cdots, n\}, \quad & -v_i^- + v_j^+ \leq b_{ij}
\end{aligned}
\tag{5.3}
$$

By transforming the optimization formulation into linear programming, we can ensure an important and nice theoretic property that every local minimum is a global minimum. Now, we can utilize many known methods and toolkits to solve it efficiently.

5.5 Experiments

We employ PyGLPK toolkit (a Python module encapsulating the functionality of the GNU Linear Programming Kit) to solve the linear programming problem we formulated. All experiments are performed when setting the number of topics to be five, and all λs to be the same.

Topics	# of Posts per User	# of Posts per Thread	# of ReplyTo
abortion	3.19	59.4	27
healthcare reform	3.85	64.6	29.2
illegal immigrants	2.94	61.4	24.6
iraq war	3.31	64.8	26.4
president barack obama	3.22	61.8	24.8

Table 5.1: Basic Statistics of Data Sets

5.5.1 Data sets Description

Discovering opinion networks is a new problem and there is no existing data set that can be used for evaluation. We create our own data sets from an online military forum[1]. We crawl all 43,483 threads of discussions, containing 1,343,427 posts, from the "Hot Topics & Current Events" category. Then, we narrow down to five popular and controversial topics: abortion, healthcare reform, illegal immigrants, iraq war, and president barack obama, so that it would be easier for the human judges to annotate. Using the keywords for each topic, we use retrieval techniques to select five threads, where each thread has between 40 and 90 posts. The basic statistics are summarized in Table 5.1.

5.5.2 Human annotation

In order to get the ground truth data for evaluation purposes, we have two efforts:

- First, we set up our own interface for human annotation and ask our fellow colleagues to read post content carefully and label each as "For", "Against" or "Not Sure" about the given topic. From the limited response, we get 230 posts with labels agreed by at least two people, where 26% as "For", 41% as "Against" and 33% as "Not Sure". Also, out of the disagreement among our annotators, the true disagreement (where one label as "For" while the other label as "Against") rate is only 12.31%. This shows that the task is reasonabe for human judges.

[1] `forums.military.com`

- The first step of human annotation shows that the defined task can be performed by human users at a reasonably good accuracy/agreement. However, the size of the ground truth we get is too small for meaningful quantitative evaluation. So, we further utilize crowd sourcing service through CrowdFlower[2]. More specifically, we submit all 1584 posts from the 25 threads of 5 topics into their online system, requiring each post to be labeled by at least three annotators. In order to better control the quality of their annotation, we also define a set of "gold" which are from the 230 posts with agreed labels from our first round of annotation by our colleagues. In cases when there are strong disagreement between the crowd sourcing annotators and our colleagues, we make the final decision by ourselves.

  The annotation results from CrowdFlower basically follows the statistics of the first round annotation: 30% as "For", 43% as "Against" and 26% as "Not Sure", suggesting its trustworthiness. In the following evaluation, we will use this bigger set of annotation results as the ground truth.

5.5.3 Methods for Comparison

In addition to the Linear Programming (LP) method we have proposed, we also include three other methods for comparison purpose:

1. **UserClustering:** We consider each user as a bag of words by concatenating all her posts. We build similarity graph among users from cosine similarity and then apply graph partition based clustering of two groups of users (using the CLUTO toolkit[3]). This baseline does not use the information around each post.

2. **SentiWordNet:** We first perform sentiment tagging of each post by averaging the sentiment score of each word in SentiWordNet, which will produce an opinion score

[2] `www.crowdflower.com`
[3] `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview`

between -1 and 1 for each post. Then each user's opinion is the averaged opinion score of all her posts. We need to apply a threshold α to decide the supporting group (users with opinion scores larger than α) and the against group (users with opinion scores smaller than $-\alpha$). This baseline represents an unsupervised sentiment analysis method which exploits only the textual content.

3. **MaxCut:** The method proposed in [62] is the closest to our work. Basically, they first classify each post-post reply-to relation as agree/disagree/neutral using a dictionary-matching approach; then, user-user relation is defined as a linear combination of their post-post relations (positive if mostly disagree and negative is mostly agree); finally, a MaxCut algorithm is performed on this user-user graph. Since we do not have their algorithm implementation or their pattern dictionary, we use SentiWordNet and a minimal set of patterns (same as the input to our method) as the first step classifier. The next two steps are implemented same as what has been described in their paper.

5.5.4 Evaluation of Agree/Disagree Classification

We first evaluate the accuracy of the local classification of agree/disagree relations between posts, which are the signals to be fed into the optimization. Instead of asking human judges to annotate every pair of posts, which would involve many combinations and repeated judgments, we simply ask human judges to read each post and label it as "For", "Against" or "Not Sure". Then the ground truth post-post relation can be derived from such annotated results: given any two posts in one thread, they are in agreement if both of them are annotated as "For" or both of them are annotated as "Against"; they are in disagreement if one is annotated as "For" and the other annotated as "Against". We ignore the cases when one post is annotated as "Not Sure".

In Table 5.2, we compare the results from the first step of MaxCut and our methods of extracting matrices R (derived from reply-to relation), T (derived from topic modeling)and

A (derived from user consistency assumption). The best performing method for each metric is highlighted in bold font.

Method	Precision	Recall	F1 Measure
MaxCut	0.4732	0.0090	0.0177
$R+T+A$	0.6010	**0.1942**	**0.2936**
R	0.5582	0.0036	0.0071
T	0.5632	0.1134	0.1888
A	**0.6791**	0.0900	0.1589

Table 5.2: Accuracy of Agree/Disagree Classification

Our observation from the results include the following,

1. From the top part of the table, we can see that by combining textual information and social interaction information our method outperform the MaxCut method proposed in [62] in all metrics.

   Since MaxCut use a rule-base classifier, the coverage of the patterns is hard to guarantee especially when we want to handle very different types of forums/issues which involve different vocabulary and possibly slangs too. Our methods relies on only a handful very basic patterns of showing agreement/disagreement (six in total); instead, our classifier is learned by exploiting the forum data itself using both textual analysis and social network analysis. This kind of approach can automatically adapt to different types of topics or forums.

2. We also want to understand the contribution of each component in our method. For example, are they complementary with each other? Are they making similar or different mistakes? So, we evaluate each component of our method separately in the bottom part of the table.

   First, the A matrix performs the best in precision, indicating that author consistency assumption is most accurate in identifying post-post relation.

Second, the A and T matrices gives much higher recall than the R matrix. This is because that there are only a subset of posts have reply-to information, out of which we only output the most confident cases in our bootstrapping approach when producing R. In comparison, A matrix relies on multiple posts written by the same user, which is much richer than reply-to information; T matrix only depends on the post content, thus applies to all posts, generating much higher recall.

Finally, the recall of $R+T+A$ is almost the same as the sum of recall of the three matrices independently. This suggests that these three matrices are mostly not overlapping with each other, each providing complementary information. Thus, by combining them together, we can get the best performance.

Now that we have shown that our methods discover more accurate relations *between posts*, in the next set of evaluation we will further test the performance of using these post-post relations to discover opposing opinion network.

5.5.5 Evaluation of Opposing Opinion Network

Once we have the opinion in each post predicted, we aggregate the opinions to the user level (by averaging the opinions of all her posts). For ground truth, we only take the most confident ones: users are in the ground truth supporting group only if each has composed at least two posts and the aggregated opinion score is larger than 0.5; similarly, users are in the ground truth against group only if each has composed at least two posts and the aggregated opinion score is smaller than -0.5. This results in 57 users in the supporting group and 78 in the against group. Then, we only consider the two classes of "For" and "Against" (which are the interesting ones), and evaluate accuracy of the algorithm prediction. Note that the ground truth is only a subset of users that we have confident human labels, so we can only evaluate the accuracy of each algorithm on this subset. We also evaluate mean squared error (MSE) which is more accurate measure capturing the intuition that predicting a supporting

Method	Accuracy (Supporting)	Accuracy (Against)	Accuracy (Both)	MSE
UserClustering	0.6250	0.4615	0.5299	0.4535
MaxCut	0.6964	0.3590	0.5000	0.4703
SentiWordNet (cutoff=0)	0.5357	0.2318	0.4851	0.4376
LP (cutoff=0)	0.5769	**0.5844**	**0.5814**	**0.4352**
SentiWordNet (MaxCut Partition)	0.6964	0.4103	0.5299	0.4631
LP (MaxCut Partition)	**0.7500**	0.3896	0.5349	0.4522

Table 5.3: Accuracy of User Opinion Prediction

user to an against group is much worse than predicting her opinion as "not sure".

In Table 5.3, we compare three baselines with our Linear Programming (LP) method. Both SentiWordNet and LP predict an opinion score for each user, so we can use a default cutoff of zero to decide the partition of supporting group and against group. We evaluate the accuracy of the supporting group and against group separately and jointly, as shown in the table columns. We also highlight the best method in bold font for each measure.

We can see that

1. UserClustering is not performing well, which shows that treating each user as a big document and ignoring the relations among posts is not effective.

2. MaxCut is doing even worse than UserClustering. The big difference between the accuracy in two groups suggests that the MaxCut partition of users may be unbalanced.

3. SentiWordNet (cutoff=0) gives the lowest accuracy, suggesting that relying solely on the content and sentiment lexicon matching is not very trustable.

4. our LP (cutoff=0) method outperforms UserClustering and MaxCut in both accuracy and MSE. It means that the score assignment from our optimization formulation is meaningful and also provides flexibility to partition the users. In comparison, UserClustering and MaxCut provide a hard partition that is not as accurate. Our LP method also outperforms SentiWordNet in every measure. Since we use SentiWordNet as one term in the objective function, this shows that the other terms capturing

agreement/disagreement further help adjusting the opinion scores more accurately.

5. In order to account for the possible bias coming from the partition size, we further evaluate SentiWordNet and LP using the same partition size as MaxCut. (shown in the bottom part of the table) More specifically, we rank the users by the opinoin scores output by SentiWordNet or LP, and then partition into supporting/against group is done so that the number of supporting/against users is the same as MaxCut output. As can be seen in the table, both methods ourperform MaxCut, showing that the output opinion socres are not only more flexible than a hard partition but also more accurate.

5.5.6 Application I: Measuring Topic Correlation

Once we infer the latent opinion network for each different topic, we can now measure the correlation between topics based on the similarity of their corresponding opinion networks. This is a measure of topical similarity at the semantic level going beyond textual similarity. For example, topic A and topic B may share very little vocabulary, but using our measure we can find their hidden correlation or similarity because those people supporting/against topic A are also supporting/against topic B.

To demonstrate this application, we first represent each topic t as a vector $OpNet_t$ of size m (the number of users involved); the value of the vector element is between -1 and 1, basically the user's opinion score toward this topic. Then we measure the topic correlation of two topics t_1 and t_2 by the cosine similarity of their corresponding vector representation:

$$Topic\ Correlation = cosine(OpNet_{t_1}, OpNet_{t_2})$$

To test this measure, we apply it to each pair of the five topics in Table 5.1. We rank the topic pairs based on Topic Correlation in Table 5.4. We can see that the most positively correlated topics are "healthcare reform" and "abortion" and the most negatively correlated

Topic 1	Topic 2	Topic Correlation
healthcare reform	abortion	0.0806
president barack obama	healthcare reform	0.0588
iraq war	abortion	0.0421
president barack obama	abortion	0.0356
president barack obama	illegal immigrants	0.0080
illegal immigrants	healthcare reform	-0.0015
president barack obama	iraq war	-0.0094
iraq war	illegal immigrants	-0.0101
illegal immigrants	abortion	-0.0162
iraq war	healthcare reform	-0.0571

Table 5.4: Ranking of Topic Correlation

topics are "Iraq war" and "healthcare reform". This is consistent with our knowledge that most Democrats are supporting health care reform and are pro-choice in the abortion issue while against Iraq war.

5.5.7 Application II: Measuring User Similarity

Similarly, with the latent opinion network inferred, we can also measure the correlation between users based on the similarity of their opinions across different topics.

We represent each user u as a vector Op_u of size k (the number of topics). Then we measure the user opinion similarity of two users u_1 and $u_2$by the cosine similarity of their corresponding vector representation:

$$User\ Similarity = cosine(Op_{u_1}, Op_{u_2})$$

We test this user opinion similarity measure on the five topics. Due to the limit of space, we only look at the user pair with the largest similarity (User X and User Y) and the user pair with the least similarity (User Y and User Z). Real user names have been anonymized.

In order to qualitatively validate the results, we check the original content posted by the three users. For example, User X replied "If you folks are so unhappy I hear Canada is

ni ce this time of year." quoting some previous post – "It is very damning. Obama didn apos;t report the solicitation. That is a crime in itself."; and User Y posted "this whole birth certificate thing is a total non issue, made up by people with WAY TOO MUCH time on their hands." suggesting his support for President Barack Obama too.

Another example is that User Y replied "A very stupid one." to "What kind of question is this?" which was from User B in Figure 5.1 while User Z said that "So a mother can kill an unborn child because of economic reasons, so should it be legal for parents to kill their children after they are born because of financial hardship. That answer is obviously no, so why is alright to kill an unborn child because of money issues.". Clearly, they are holding opposing views on the abortion issue.

Using this new measure of user similarity at the semantic level, we can support interesting applications such as recommending users with the most similar opinions or those with the most opposite opinions. This would enhance user experience in forum participation and potentially bring in social components to forums.

5.6 Conclusions and Future Work

We define the novel problem of discovering opinion networks, which are essentially latent social networks based on user sentiment/position. We study a special case of opposing opinion networks of users and propose method to discover them in an unsupervised way from forums. In particular, we analyze signals from both textual content (e.g. post content) and social interactions (e.g. who talks to whom) and design an optimization formulation to combine all the signals in a unified manner. We test the effectiveness of the proposed method using a manually annotated forum data on five controversial topics. Experimental results show that the proposed optimization method outperforms several baselines and existing approaches, demonstrating the power of combining both text analysis and social network analysis in discovering opinion networks. We also demonstrate two interesting applications

of the discovered opinion networks in finding semantically similar topics and recommending similar-minded users, respectively.

Our work opens up a novel direction in text mining where the focus is on analyzing latent user behavior and social structure behind text. There are many interesting future research directions to further explore. For example, we may further infer more specific relations among users from the opinion network, e.g. friends, enemies, followers, etc. We are also interested in enhancing our current method by learning the λ weights automatically. The current way of setting all λ weights to be the same may not be the optimal way. An iterative approach may be promising in setting the optimal λ weights automatically: first infer an opposing network first so that we can measure topic similarity, then use the topic similarity as external guidance to adjust the weights, then iterative until convergence.

Chapter 6

Opinion Quality Prediction

The rapid growth of opinion data in Web 2.0 applications comes at the price of wide variance of the quality which may compromise the usefulness of the information. Thus, automatically and accurately assessing opinion quality is a pressing challenge for opinion integration and analysis.

6.1 Introduction

Web 2.0 has empowered users to actively interact with each other, forming social networks around mutually interesting information and publishing large amounts of useful user-generated content online. Unfortunately, the abundance of user-generated content comes at a price. For every interesting opinion, or helpful review, there are also large amounts of spam content, unhelpful opinions, as well as highly subjective and misleading information. Sifting through large quantities of reviews to identify high quality and useful information is a tedious, error-prone process. It is thus highly desirable to develop reliable methods to assess the quality of reviews automatically. Robust and reliable review quality prediction will enable sites to surface high-quality reviews to users while benefiting other important popular applications such as sentiment extraction and review summarization [36, 35], by providing high-quality content on which to operate.

Automatic review quality prediction is useful even for sites providing a mechanism where users can evaluate or rate the helpfulness of a review (e.g. `Amazon.com` and `Epinions.com`). Not all reviews receive the same helpfulness evaluation [43]. There is a rich-get-richer

effect [48] where the top reviews accumulate more and more ratings, while more recent reviews are rarely read and thus not rated. Furthermore, such helpfulness evaluation is available only within a specific Web site, and is not comparable across different sources. However, it would be more useful for users if reviews from different sources for the same item could be aggregated and rated automatically on the same scale. This need is addressed by a number of increasingly popular aggregation sites such as `Wize.com`. For these sites, automatic review rating is essential in order to meaningfully present the collected reviews.

Most previous work [91, 43, 48, 25, 49, 80] attempts to solve the problem of review evaluation by treating each review as a stand-alone text document, extracting features from the text and learning a function based on these features for predicting review quality. However, in addition to textual content, there is much more information available that is useful for this task. Online reviews are produced by identifiable authors (reviewers) who interact with one another to form social networks. The history of reviewers and their social network interactions provide a *social context* for the reviews. In our approach, we mine combined textual, and social context information to evaluate the quality of individual reviewers and to assess the quality of the reviews.

In this chapter, we investigate how the social context of reviews can help enhance the accuracy of a text-based quality predictor. To the best of our knowledge, this is the first time that textual, author and social network information are combined for assessing review quality. Expressed very generally, our idea is that social context reveals a lot about the quality of reviewers, which in turn affects the quality of the reviews. We formulate hypotheses that capture this intuition and then mathematically model these hypotheses by developing regularization constraints which augment text-based review quality prediction. The resulting quality predictor is formulated into a well-formed convex optimization problem with efficient solution. The proposed regularization framework falls under the category of semi-supervised learning, making use of a small amount of labeled data as well as a large amount of unlabeled data. It also has the advantage that the learned predictor is applicable to any review,

even reviews from different sources or reviews for which the reviewer's social context is not available. Finally, we experiment with real review data from an online commerce portal. We test our hypotheses and show that they hold for all three categories of data we consider. We then experimentally demonstrate that our novel regularization methods that combine social context with text information can lead to improved accuracy of review quality prediction, especially when the available training data is sparse.

6.2 Problem Definition

A review system consists of three sets of three different types of entities: a set $I = \{i_1, ..., i_N\}$ of N *items* (products, events, or services); a set $R = \{r_1, ..., r_n\}$ of n *reviews* over these items; and a set $U = \{u_1, ..., u_m\}$ of m *reviewers* (or *users*) that have authored these reviews. Each entity has a set of attributes T associated with it. For an item i or a user u, T_i and T_u are sets of attribute-value pairs describing the item and the user respectively while for a review r, T_r is the text of the review. We are also given relationships between these sets of entities. There is a function $M : R \to I$ that maps each review r to a unique item $i_r = M(r)$; an authorship function $A : R \to U$, that maps each review r to a unique reviewer $u_r = A(r)$; and a relation $S \subset U \times U$ that defines the social network relationships between users.

Since each review is associated with a unique item, we omit the set I, unless necessary, and assume all information about the item i_r (item identifier and attributes) is included as part of the attributes T_r of review r. We also model the social network relation as a *directed* graph $G_S = (U, S)$ with adjacency matrix $\mathbf{S}$, where $\mathbf{S}_{uv} = 1$ if there is a link or edge from u to v and zero otherwise. We assume that the links between users in the social network capture semantics of trust and friendship: the meaning of user u linking to user v is that u values the opinions of user v as a reviewer.

The information about the authors of the reviews along with the social network of the reviewers places the reviews within a *social context*. More formally we have the following

definition.

Social Context Given a set of reviews R, we define the *social context* of the set R as the triple $C(R) = \langle U, A, S \rangle$, of the set of reviewers U, the authorship function A, and the social network relation S.

The set of reviews R contains both *labeled* (R_L) and *unlabeled* (R_U) reviews. For each review $r_i \in R_L$ in the labeled subset of reviews we observe a numeric value q_i that captures the true quality and helpfulness of the review. We use $L = \{(r_i, q_i)\}$, to denote the set of review-quality pairs. Such quality values can be obtained through manual labeling or through feedback mechanisms in place for some online portals.

Given the input data $\{R_L \cup R_U, C(R), L\}$, we want to learn a *quality predictor* Q that, for a review r, predicts the quality of the review. A review r is represented as an f-dimensional real vector $\mathbf{r}$ over a feature space F constructed from the information in R and $C(R)$. So the quality predictor is a function $Q : \mathbb{R}^f \rightarrow \mathbb{R}$ that maps a review feature vector to a numerical quality value.

Previous work has used the information in $\{R_L, L\}$ for learning a quality predictor, based mostly on different kinds of textual features. In this work, we investigate how to enhance the quality predictor function Q using the social context $C(R)$ of the reviews in addition to the information in $\{R_L, L\}$. Our exploration for the prediction function Q takes the following steps. First we construct a text-based baseline predictor that makes use of only the information in $\{R_L, L\}$. Then we enhance this predictor by adding social context features that we extract from $C(R_L)$. In the last step, which is the focus of this work, we propose a novel semi-supervised technique that makes use of the labeled data $\{R_L, L\}$, the unlabeled data R_U, and the social context information $C(R)$ for both labeled and unlabeled data.

6.3 Text-Based Quality Prediction

The text of a review provides rich information about its quality. In this section, we build a baseline supervised predictor that makes use of a variety of textual features as detailed in the top part of Table 6.1. We group the features into four different types.

1. **Text-statistics features**: This category includes features that are based on aggregate statistics over the text, such as the length of the review, the average length of a sentence, or the richness of the vocabulary.

2. **Syntactic Features**: This category includes features that take into account the Part-Of-Speech (POS) tagging of the words in the text. We collect statistics based on the POS tags to create features such as percentage of nouns, adjectives, punctuations, etc.

3. **Conformity features**: This category compares a review r with other reviews by looking at the KL-divergence between the unigram language model T_r of the review r for item i, and the unigram model $\overline{T}_i$ of an "average" review that contains the text of all reviews for item i. This feature is used to measure how much the review conforms to the average and is defined as $D_{KL}(T_r||\overline{T}_i) = \sum_w T_r(w) \log(T_r(w)/\overline{T}_i(w))$ where w takes values over the tokens of the unigram models.

4. **Sentiment features**:This category considers features that take into account the positive or negative sentiment of words in the review. The occurrence of such words is a good indication about the strength of the opinion of the reviewer.

With this feature set F, we can now represent each review r as an f-dimensional vector $\mathbf{r}$. Given the labeled data in $\{R_L, L\}$, we want to learn a function $Q : \mathbb{R}^f \rightarrow \mathbb{R}$ that for a review $\mathbf{r}_i$ it predicts a numerical value $\hat{q}_i$ as its quality. We formulate the problem as a linear regression problem, where the function Q is defined as a linear combination of the features in F. More formally, the function Q is fully defined by an f-dimensional column

Feature Name	Type	Feature Description
TEXT FEATURES		
NumToken	Text-Stat	Total number of tokens.
NumSent	Text-Stat	Total number of sentences.
UniqWordRatio	Text-Stat	Ratio of unique words
SentLen	Text-Stat	Average sentence length.
CapRatio	Text-Stat	Ratio of capitalized sentences.
POS:NN	Syntactic	Ratio of nouns.
POS:ADJ	Syntactic	Ratio of adjectives.
POS:COMP	Syntactic	Ratio of comparatives.
POS:V:	Syntactic	Ratio of verbs.
POS:RB	Syntactic	Ratio of adverbs.
POS:FW	Syntactic	Ratio of foreign words.
POS:SYM	Syntactic	Ratio of symbols.
POS:CD	Syntactic	Ratio of numbers.
POS:PP	Syntactic	Ratio of punctuation symbols.
KLall	Conformity	KL div $D_{KL}(T_r \| \overline{T}_i)$
PosSEN	Sentiment	Ratio of positive sentiment words.
NegSEN	Sentiment	Ratio of negative sentiment words.
SOCIAL NETWORK FEATURES		
ReviewNum	Author	Num. of past reviews by the author.
AvgRating	Author	Past average rating for the author.
In-Degree	SocialNetwork	In-degree of the author.
Out-Degree	SocialNetwork	Out-degree of the author.
PageRank	SocialNetwork	PageRank score of the author.

Table 6.1: Textual Features and Social Context Features

weight vector $\mathbf{w}$, such that $Q(\mathbf{r}) = \mathbf{w}^T\mathbf{r}$, where $\mathbf{w}^T$ denotes the transpose of the vector. In the following, since Q is uniquely determined the by weight vector $\mathbf{w}$ and vice versa, we will use Q and $\mathbf{w}$ interchangeably. Our goal is to find the f-dimensional weight vector $\hat{\mathbf{w}}$ that minimizes the objective function:

$$\Omega(\mathbf{w}) = \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\mathcal{L}(\mathbf{w}^T\mathbf{r}_i, q_i) + \alpha\mathbf{w}^T\mathbf{w} \tag{6.1}$$

where $\mathcal{L}$ is the loss function that measures distance of the predicted quality $Q(\mathbf{r}_i) = \mathbf{w}^T\mathbf{r}_i$ of review $r_i \in R_L$ with the true quality value q_i, n_ℓ is the number of training examples, and $\alpha \geq 0$ is regularization parameter for $\mathbf{w}$. In our work, we use squared error loss (or quadratic

loss), and we minimize the function

$$\Omega_1(\mathbf{w}) = \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}(\mathbf{w}^T\mathbf{r}_i - q_i)^2 + \alpha\mathbf{w}^T\mathbf{w} \tag{6.2}$$

The closed form solution for $\hat{\mathbf{w}}$ is given by

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \Omega_1(\mathbf{w}) = (\sum_{i=1}^{n_\ell}\mathbf{r}_i\mathbf{r}_i^T + \alpha n_\ell\mathcal{I})^{-1}\sum_{i=1}^{n_\ell} q_i\mathbf{r}_i$$

where $\mathcal{I}$ is the identity matrix of size f.

Once we have learned the weight vector $\mathbf{w}$, we can apply it to any review feature vector and predict the quality of unlabeled reviews.

6.4 Incorporating Social Context

The solution we describe in Section 6.3 considers each review as a stand-alone text document. As we have discussed, in many cases we also have available the social context of the reviews, that is, additional information about the authors of the reviews, and their social network. In this section we discuss different ways of incorporating social context into the quality predictor we described in Section 6.3. Our work is based on the following two premises:

1. The quality of a review depends on the quality of the reviewer. Estimating the quality of the reviewer can help in estimating the quality of the review.

2. The quality of a reviewer depends on the quality of their peers in the social network. We can obtain information about the quality of the reviewers using information from the quality of their friends in their social network.

We investigate two different ways of incorporating the social context information into the linear quality predictor. The first is a straightforward expansion of the feature space to include features extracted from the social context. The second approach is novel in that

it defines constraints between reviews, and between reviewers, and adds regularizers to the linear regression formulation to enforce these constraints. We describe these two approaches in detail in the following sections.

6.4.1 Extracting features from social context

A straightforward use of the social context information is by extracting additional features for the quality predictor function. The social context features we consider are shown in the bottom part of Table 6.1. The features capture the engagement of the author (ReviewNum), the historical quality of the reviewer (AvgRating), and the status of the author in the social network (In/Out-Degree, PageRank).

This approach of using social context is simple and it fits directly into our existing linear regression formulation. We can still use Equation 6.2 for optimizing the function Q, which is now defined over the expanded feature set F. The disadvantage is that such information is not always available for all reviews. Consider for example, a review written anonymously, or a review by a new user with no history or social network information. Predicting using social network features is no longer applicable. Furthermore, as the dimension of features increases, the necessary amount of labeled training data to learn a good prediction function also increases.

6.4.2 Extracting constraints from social context

We now present a novel alternative use of the social context that does not rely on explicit features, but instead defines a set of constraints for the text-based predictor. These constraints define hypotheses about how reviewers behave individually or within the social network. We require that the quality predictor respects these constraints, forcing our objective function to take into account relationships between reviews, and between different reviewers.

Social Context Hypotheses

We now describe our hypotheses, and how these hypotheses can be used in enhancing the prediction of the review quality. In Section 6.5 we validate them experimentally on real-world data, and we demonstrate that they hold for all the three data sets we consider.

1. **Author Consistency Hypothesis:** The hypothesis is that reviews from the same author will be of similar quality. A reviewer that writes high quality reviews is likely to continue writing good reviews, while a reviewer with poor reviews is likely to continue writing poor reviews.

2. **Trust Consistency Hypothesis:** We make the assumption that a link from a user u_1 to a user u_2 is an explicit or implicit statement of trust. The hypothesis is that the reviewers trust other reviewers in a rational way. In this case, reviewer u_1 trusts reviewer u_2 only if the quality of reviewer u_2 is at least as high as that of reviewer u_1. Intuitively, we claim that it does not make sense for users in the social network to trust someone with quality lower than themselves.

3. **Co-Citation Consistency Hypothesis:** The hypothesis is that people are consistent in how they trust other people. So if two reviewers u_1, and u_2 are trusted by the same third reviewer u_3, then their quality should be similar.

4. **Link Consistency Hypothesis:** The hypothesis is that if two people are connected in the social network (u_1 trusts u_2, or u_2 trusts u_1, or both), then their quality should be similar. The intuition is that two users that are linked to each other in some way, are more likely to share similar characteristics than two random users. This is the weakest of the four hypotheses but we observed that it is still useful in practice.

Exploiting hypotheses for regularization

We now describe how we enforce the hypotheses defined above by designing regularizing constraints to add into the text-based linear regression defined in Section 6.3.

1. **Author Consistency**: We enforce this hypothesis by adding a regularization term into the regression model where we require that the quality of reviews from the same author is similar. Let R_u denote the set of reviews authored by reviewer u, including both labeled and unlabeled reviews. Then the objective function becomes:

$$\Omega_2(Q) = \Omega_1(Q) + \beta \sum_{u \in U} \sum_{r_i, r_j \in R_u} (Q(\mathbf{r}_i) - Q(\mathbf{r}_j))^2 \tag{6.3}$$

Minimizing the regularization constraint will force reviews of the same author u to receive similar quality values. We can formulate this as a graph regularization. The graph adjacency matrix $\mathbf{A}$ is defined as $\mathbf{A}_{ij} = 1$ if review r_i and review r_j are authored by the same reviewer, and zero otherwise. Then, Equation 6.3 becomes:

$$\begin{aligned} \Omega_2(\mathbf{w}) = & \quad \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} \left(\mathbf{w}^T \mathbf{r}_i - q_i\right)^2 + \alpha \mathbf{w}^T \mathbf{w} \\ & + \beta \sum_{i<j} \mathbf{A}_{ij} \left(\mathbf{w}^T \mathbf{r}_i - \mathbf{w}^T \mathbf{r}_j\right)^2 \end{aligned} \tag{6.4}$$

Let $\mathbf{R} = [\mathbf{r}_1, ..., \mathbf{r}_n]$ be an $f \times n$ feature-review matrix defined over *all* reviews (both labeled and unlabeled). Then the last regularization constraint of Equation 6.4 can be written as

$$\sum_{i<j} \mathbf{A}_{ij} \left(\mathbf{w}^T \mathbf{r}_i - \mathbf{w}^T \mathbf{r}_j\right)^2 = \mathbf{w}^T \mathbf{R} \Delta_A \mathbf{R}^T \mathbf{w}$$

$\Delta_{\mathbf{A}} = \mathbf{D}_{\mathbf{A}} - \mathbf{A}$ is the graph Laplacian, and $\mathbf{D}_{\mathbf{A}}$ is a diagonal matrix with $\mathbf{D}_{\mathbf{A}ii} = \sum_j \mathbf{A}_{ij}$. The new optimization problem is still convex with the closed form solu-

tion [96]:

$$\hat{\mathbf{w}} = (\sum_{i=1}^{n_\ell} \mathbf{r}_i \mathbf{r}_i^T + \alpha n_\ell \mathcal{I} + \beta n_\ell \mathbf{R} \Delta_{\mathbf{A}} \mathbf{R}^T)^{-1} \sum_{i=1}^{n_\ell} q_i \mathbf{r}_i$$

2. **Trust Consistency**: Let u be a reviewer. Given a review quality predictor function Q, we define the *reviewer* quality $\bar{Q}(u)$ as the average quality of all the reviews authored by this reviewer as it is estimated by our quality predictor. That is,

$$\bar{Q}(u) = \frac{\sum_{r \in R_u} Q(\mathbf{r})}{|R_u|} = \frac{\sum_{r \in R_u} \mathbf{w}^T \mathbf{r}_i}{|R_u|} \tag{6.5}$$

We enforce the trust consistency hypothesis by adding a regularization constraint to Equation 6.2. Let N_u denote the set of reviewers that are linked to by reviewer u. We have

$$\Omega_3(Q) = \Omega_1(Q) + \beta \sum_{u_1} \sum_{u_2 \in N_{u_1}} \left(\max\left\{0, \bar{Q}(u_1) - \bar{Q}(u_2)\right\}\right)^2$$

The regularization term is greater than zero for each pair of reviewers u_1 and u_2 where u_1 trusts u_2, but the estimated quality of u_1 is greater than that of u_2. Minimizing function Ω_3 will push such cases closer to zero, forcing the quality of a reviewer u_1 to be no more than that of u_2, and thus enforcing the trust consistency hypothesis.

Formally, for a reviewer u, let $\mathbf{h}_u$ be the n-dimensional normalized indicator vector where $\mathbf{h}_u(i) = 1/|R_u|$ if user u has written review r_i, and zero otherwise. Then we have that $\bar{Q}(u) = \mathbf{w}^T \mathbf{R} \mathbf{h}_u$. We can thus write the objective function as

$$\begin{aligned} \Omega_3(\mathbf{w}) = \quad & \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} (\mathbf{w}^T \mathbf{r}_i - q_i)^2 + \alpha \mathbf{w}^T \mathbf{w} \\ + \quad & \beta \sum_{u,v \in U} \mathbf{S}_{uv} \left(\max\left\{0, \mathbf{w}^T \mathbf{R} \mathbf{h}_u - \mathbf{w}^T \mathbf{R} \mathbf{h}_v\right\}\right)^2 \end{aligned} \tag{6.6}$$

where $\mathbf{S}$ is the social network matrix. The optimization problem is still convex, but due to the max function, no nice closed form solution exists. We can still solve it and find the global optimum by gradient descent, where the gradient of the objective

function is

$$\begin{aligned}\frac{\partial \Omega_3(\mathbf{w})}{2\partial \mathbf{w}} &= \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\mathbf{r}_i\mathbf{r}_i^T\mathbf{w} - \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\mathbf{r}_i q_i + \alpha\mathbf{w} \\ &+ \beta \sum_{\substack{u,v,\\ \mathbf{w}^T\mathbf{R}(\mathbf{h}_u-\mathbf{h}_v)>0}} \mathbf{S}_{uv}\mathbf{R}(\mathbf{h}_u-\mathbf{h}_v)(\mathbf{h}_u-\mathbf{h}_v)^T\mathbf{R}^T\mathbf{w}\end{aligned}$$

Let $\mathbf{H} = [\mathbf{h}_1, ..., \mathbf{h}_m]$ be an $n \times m$ matrix defined over all reviewers and $\mathbf{Z}$ be a new matrix such that

$$\mathbf{Z}_{uv} = \begin{cases} S_{uv} & \text{if } \left[\text{diag}(\mathbf{w}^T\mathbf{RH})\mathbf{S} - \mathbf{S}\ \text{diag}(\mathbf{w}^T\mathbf{RH})\right]_{uv} > 0 \\ 0 & \text{otherwise} \end{cases}$$

Now we can rewrite the gradient as

$$\frac{\partial \Omega_3(\mathbf{w})}{2\partial \mathbf{w}} = \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\mathbf{r}_i\mathbf{r}_i^T\mathbf{w} - \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\mathbf{r}_i q_i + \alpha\mathbf{w} + \beta\mathbf{RH}\Delta_{\mathbf{Z}}\mathbf{H}^T\mathbf{R}^T\mathbf{w}$$

where $\Delta_{\mathbf{Z}} = D_{\mathbf{Z}} + D_{\mathbf{Z}^T} - \mathbf{Z} - \mathbf{Z}^T$ can be thought of the graph Laplacian generalized for directed graphs with $D_{\mathbf{Z}}$ and $D_{\mathbf{Z}^T}$ the diagonal matrices of the row, and column sums of $\mathbf{Z}$ respectively.

3. **Co-Citation Consistency**: We enforce this hypothesis by adding a regularization term into the regression model, where we require that the quality of reviews authored by two co-cited reviewers is similar. Then, the objective function (Equation 6.2) becomes:

$$\Omega_4(Q) = \Omega_1(Q) + \beta \sum_{u\in U}\sum_{x,y\in N_u}\left(\bar{Q}(x) - \bar{Q}(y)\right)^2$$

Minimizing function Ω_4 will cause the quality difference of reviewers x and y to be pushed closer to zero, making them more similar.

We can again formulate these constraints as a graph regularizaton. Let $\mathbf{C}$ be the co-citation graph adjacency matrix, where $\mathbf{C}_{ij} = 1$ if two reviewers u_i and u_j are both trusted by at least one other reviewer u. Using the same definition of matrix $\mathbf{R}$ and vector $\mathbf{h}_u$ as for trust consistency, the objective function now becomes

$$\begin{aligned} \Omega_4(\mathbf{w}) &= \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\left(\mathbf{w}^T\mathbf{r}_i - q_i\right)^2 + \alpha\mathbf{w}^T\mathbf{w} \\ &+ \beta\sum_{i<j}\mathbf{C}_{ij}\left(\mathbf{w}^T\mathbf{R}\mathbf{h}_i - \mathbf{w}^T\mathbf{R}\mathbf{h}_j\right)^2 \end{aligned} \tag{6.7}$$

Let $\Delta_{\mathbf{C}}$ be the Laplacian of graph $\mathbf{C}$. The closed form solution is

$$\hat{\mathbf{w}} = \left(\sum_{i=1}^{n_\ell}\mathbf{r}_i\mathbf{r}_i^T + \alpha n_\ell\mathcal{I} + \beta n_\ell\mathbf{R}\mathbf{H}\Delta_{\mathbf{C}}\mathbf{H}^T\mathbf{R}^T\right)^{-1}\sum_{i=1}^{n_\ell}\mathbf{r}_i q_i$$

4. **Link Consistency**: The regularization for this hypothesis is very similar to the one for the co-citation consistency. We treat the trust network as an undirected graph. Let $\mathbf{B}$ be the corresponding matrix, where $\mathbf{B}_{ij} = 1$ if $\mathbf{S}_{ij} = 1$ or $\mathbf{S}_{ji} = 1$. Our objective function now becomes

$$\begin{aligned} \Omega_5(\mathbf{w}) = & \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\left(\mathbf{w}^T\mathbf{r}_i - q_i\right)^2 + \alpha\mathbf{w}^T\mathbf{w} \\ &+ \beta\sum_{i<j}\mathbf{B}_{ij}\left(\mathbf{w}^T\mathbf{R}\mathbf{h}_i - \mathbf{w}^T\mathbf{R}\mathbf{h}_j\right)^2 \end{aligned} \tag{6.8}$$

with a similar closed form solution

$$\hat{\mathbf{w}} = \left(\sum_{i=1}^{n_\ell}\mathbf{r}_i\mathbf{r}_i^T + \alpha n_\ell\mathcal{I} + \beta n_\ell\mathbf{R}\mathbf{H}\Delta_{\mathbf{B}}\mathbf{H}^T\mathbf{R}^T\right)^{-1}\sum_{i=1}^{n_\ell}\mathbf{r}_i q_i$$

In all these cases, β is a weight on the added regularization term which defines a trade-off between the mean squared error loss and the regularization constraint in the final objective function.

Adding the regularization makes our problem a *semi-supervised* learning problem. That is, our algorithms operate on both the labeled and the unlabeled data. Although, only the labels of the labeled data are known to the algorithm, the unlabeled data are also used for optimizing the regularized regression functions. This gives considerable more flexibility to the algorithm, since it is able to operate even with little labeled data by making use of the unlabeled data and the constraints defined by the social context. Furthermore, through regularization the signal from the social context is incorporated into the textual features. The resulting predictor function operates only on textual features, so it can be applied even in the case where there is no social context.

6.5 Experiments

In this section, we present the experimental evaluation of our techniques. For our experiments we use product reviews obtained from a real online commerce portal. We begin by describing the characteristics and preprocessing of our data sets. Then, we test the hypotheses we proposed in Section 6.4.2 on these real-world datasets. Finally, we evaluate the prediction performance of different methods and conduct some analysis.

6.5.1 Data Sets

Our experiments employ the data from Ciao UK[1], a community review web site. In Ciao, people not only write critical reviews for all kinds of products and services, but also rate the reviews written by others. Furthermore, people can add members to their network of trusted members or "Circle of Trust", if they find their reviews consistently interesting and helpful.

We collected reviews, reviewers, and ratings up to May, 2009 for all products in three categories: Cellphones, Beauty, and Digital Cameras (DC). We use the average rating of the

[1] `http://www.ciao.co.uk/`

	Cellphone	Beauty	Digital Camera
PRUNING SETTINGS			
min # of ratings/ review	5	5	5
min # of reviews/reviewer	2	2	1
min # of trust links/reviewer	1	1	0
min # of reviews/ product	5	10	5
STATISTICS			
# of reviews	1943	4849	3697
# of reviewers	881	1709	3465
# of products	158	308	380
# of links in Trust	2905	20374	3894
# of links in Link	4644	32104	6022
# of links in Cocitation	13678	188610	22136
Trust graph density	0.0075	0.0140	0.0006
Link graph density	0.0120	0.0220	0.0010
Cociation graph density	0.0353	0.1292	0.0037
Avg # of reviews/reviewer	2.2054	2.8373	1.0670
Ratio of Reciprocal links	0.4014	0.4243	0.4535
Clustering coefficient	0.2286	0.3072	0.2523
CHARACTERISTICS			
Social Context	rich	rich	sparse
Quality Distribution	balanced	skewed	balanced

Table 6.2: Data Pruning Settings, Statistics, and Characteristics

reviews (a real value between 0 and 5) as our gold standard of review quality. In order for the gold standard to be robust and resistant to outlier raters, we use only reviews with at least five ratings from different raters. We then apply some further pruning by imposing the conditions shown in the top part of Table 6.2. The purpose of the pruning is to obtain a dataset that is both large enough and has sufficient social context information. Because we need some information about reviewers' history in order to test our Reviewer Consistency hypothesis, we require reviewers for Cellphone and Beauty to have at least two reviews each. We also require reviewers to be part of the trust social network (with at least one link in the social network), in order to test our hypotheses and methods based on social networks. Finally, we require for each product to have some representation in the dataset, that is, a sufficiently large number of reviews. The pruning thresholds are selected per category, so as to obtain sufficient volume of data. For the Digital Cameras category, this results in a

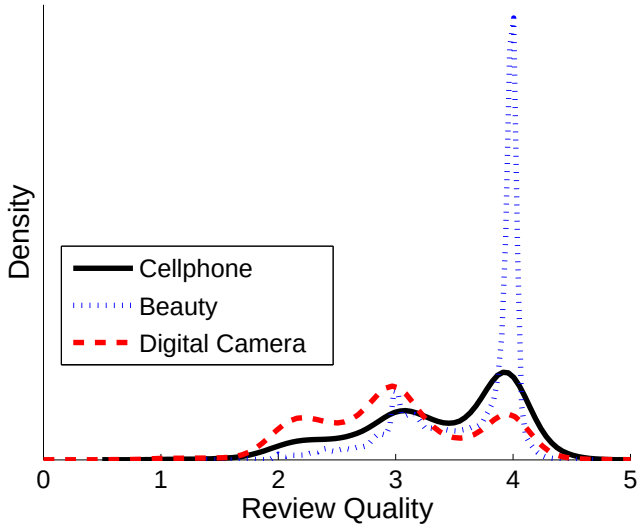


Figure 6.1: Density Estimate of Gold Standard Review Quality.

minimum amount of pruning. Although DC reviews do not contain much social context information, we still include them here for comparison and generality purposes.

From the statistics in Table 6.2, we can see that Cellphone and Beauty reviews contain more rich social context information than DC reviews in the sense that the average number of reviews per reviewer is more than twice that for Digital Cameras, and the link density (defined as $D = \frac{2|E|}{|V|(|V|-1)}$ for a graph with vertices V and edges E) is more than 10 times that of Digital Cameras. We also plot the Kernel-smoothing density estimate (pdf) of the samples q_i (the gold standard review quality) in Figure 6.1. The distributions of q_i for the three categories are quite different. Beauty reviews are highly concentrated at rating 4, while Cellphone and DC reviews have a more balanced distribution of quality. We summarize the characteristics of the three data sets in the bottom of Table 6.2.

6.5.2 Consistency Hypotheses Testing

Before evaluating the prediction performance of different algorithms, we first validate our four consistency hypotheses over our data sets.

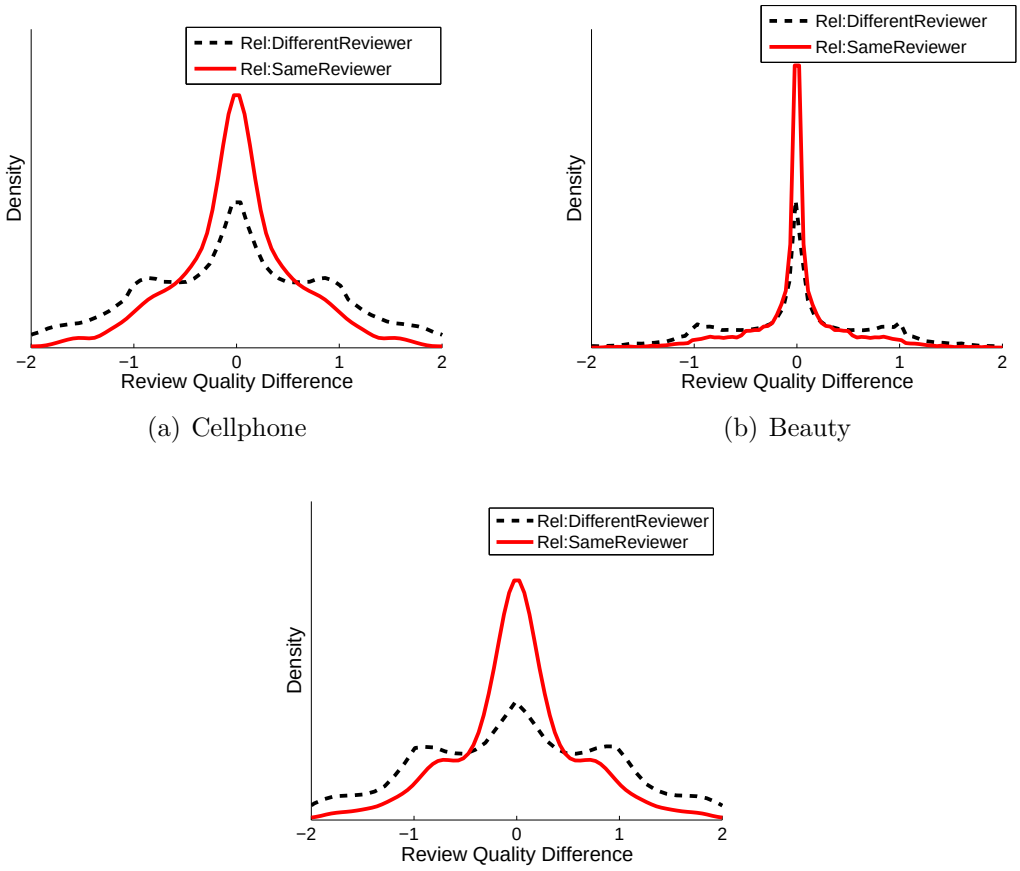
Standard Deviation	**Cellphone**	**Beauty**	**Digital Camera**
Rel:DifferentReviewer	0.9187	0.7017	0.9571
Rel:SameReviewer	0.5937	0.4518	0.6176
p-value	1.37E-48*	1.57E-287*	3.12E-11*

Table 6.3: Statistics of Review Quality Difference to Support Reviewer Consistency Hypothesis

Author Consistency Hypothesis

For each dataset, we consider all n^2 pairs of reviews (r_i, r_j), and we divide them into two disjoint groups: **Rel:DifferentReviewer** if r_i and r_j are authored by different reviewers, i.e., $u_i \neq u_j$, and **Rel:SameReviewer** if $u_i = u_j$. In each group, for each pair (r_i, r_j) we compute the difference in quality, $dq_{ij} = q_i - q_j$, of the two reviews. Since for each value dq_{ij} we also include value $dq_{ji} = -dq_{ij}$ the mean value of dq_{ij} for both groups is zero. We are interested in the standard deviation, std(dq_{ij}), that captures how much variability there is in the difference of quality between reviews for the two groups. Table 6.3 shows the results for the different datasets. For a visual comparison, in Figure 6.2 we also plot the Kernel-smoothing density estimates of the two groups.

We observe that the standard deviation of the quality difference of two reviews by the same author is much lower than that of two reviews from different authors. This indicates that reviewers are, to some extent, consistent in the quality of reviews they write. The figures also clearly indicate that the density curve for Rel:SameReviewer is more concentrated around zero than Rel:DifferentReviewer for all three categories. Moreover, two-sample Kolmogorov-Smirnov (KS) test of the samples in the two groups indicates that the difference of the two groups is statistically significant. The p-values are shown in the last row of Table 6.3. The star next to the p-value means there is strong evidence ($p < 0.01$) that the two samples come from different distributions.



(a) Cellphone

(b) Beauty

(c) Digital Cameras

Figure 6.2: Density Estimates of Review Quality Difference.

Social Network Consistency Hypotheses

In order to test the three social network consistency hypotheses, namely Trust Consistency, Co-Citation Consistency and Link Consistency, we look at the empirical distribution of $d\bar{Q}^*_{ij} = \bar{Q}^*(u_i) - \bar{Q}^*(u_j)$, i.e., the difference in quality of two reviewers, where, similar to Equation 6.5

$$\bar{Q}^*(u) = \frac{\sum_{r_i \in R_u} q_i}{|R_u|} \tag{6.9}$$

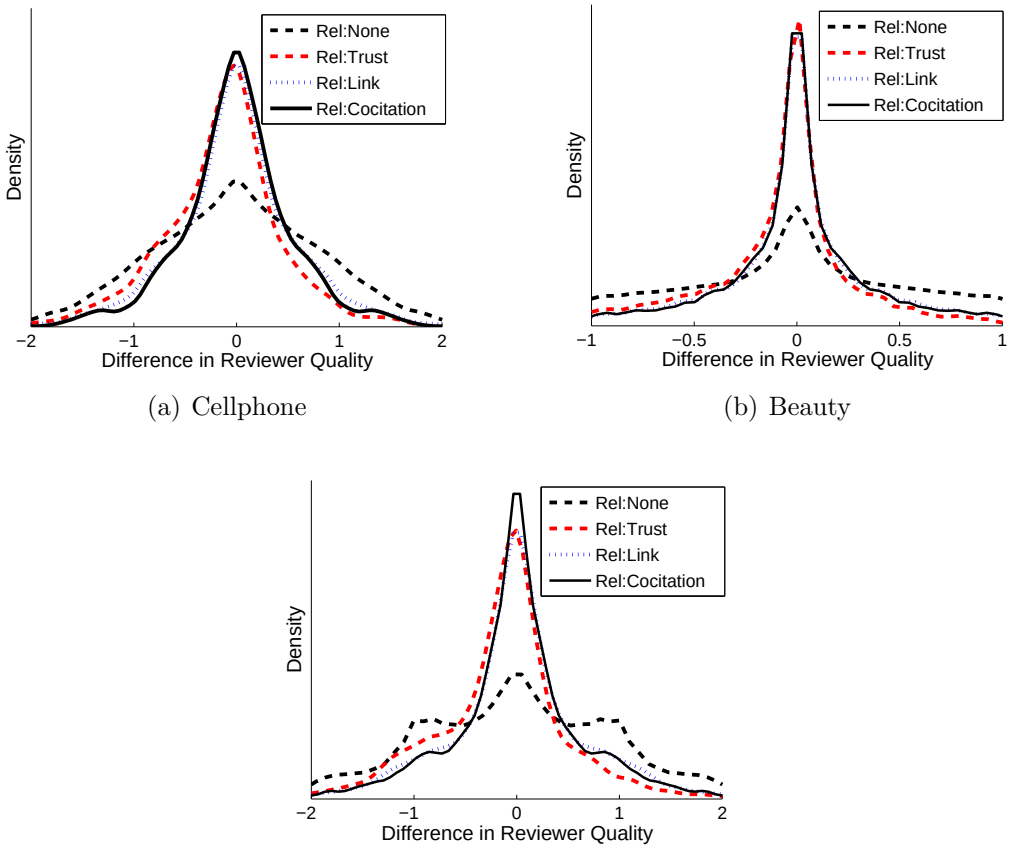
is defined as the average quality of the reviews written by u in our dataset, but using gold standard quality. Again, we group the pairs of reviewers (u_i, u_j) into the the following sets depending on the relationship between the two reviewers.

Rel:None: User u_i is not linked to user u_j, i.e., $\mathbf{B}_{ij} = 0$.

Rel:Trust: User u_i trusts user u_j, i.e., $\mathbf{S}_{ij} = 1$.

Rel:Cocitation: Users u_i and u_j are trusted by at least one other reviewer u_3, i.e., $\mathbf{C}_{ij} = 1$.

Rel:Link: User u_i trusts user u_j, or u_j trusts u_i, i.e., $\mathbf{B}_{ij} = 1$.



(a) Cellphone

(b) Beauty

(c) Digital Cameras

Figure 6.3: Density Estimates of Reviewer Quality Difference.

In Figure 6.3, we plot the Kernel-smoothing density estimate of the $d\bar{Q}^*_{ij}$ values for the four different sets of pairs, for the three categories. We further show in Table 6.4 the moments (mean and variance) of the four density estimates and p-values of the KS-test between pairs of density estimates.

The first observation is that the distribution of Rel:Trust is skewed towards the negative with a negative mean. This supports the Trust Consistency Hypothesis that when u_i trusts u_j, the quality of u_i is usually lower than that of u_j, i.e., $\bar{Q}^*(u_i) - \bar{Q}^*(u_j) < 0$. The remaining

Cellphone				
p-value	Rel:None	Rel:Trust	Rel:Link	Rel:Cocitation
Rel:None	-	3.20E-82*	4.53E-44*	6.12E-177*
Rel:Trust	-	-	3.44E-16*	6.89E-22*
Rel:Link	-	-	-	0.0657
Moments	Rel:None	Rel:Trust	Rel:Link	Rel:Cocitation
Mean	0.0000	-0.1376	0.0000	0.0000
Variance	0.6727	0.3255	0.3485	0.2914
Beauty				
p-value	Rel:None	Rel:Trust	Rel:Link	Rel:Cocitation
Rel:None	-	0.00E+00*	0.00E+00*	0.00E+00*
Rel:Trust	-	-	3.83E-59*	3.75E-101*
Rel:Link	-	-	-	0.3003
Moments	Rel:None	Rel:Trust	Rel:Link	Rel:Cocitation
Mean	0.0000	-0.0824	0.0000	0.0000
Variance	0.4331	0.1806	0.1907	0.1903
Digital Camera				
p-value	Rel:None	Rel:Trust	Rel:Link	Rel:Cocitation
Rel:None	-	1.76E-135*	2.14E-87*	0.00E+00*
Rel:Trust	-	-	1.46E-21*	2.10E-34*
Rel:Link	-	-	-	0.3052
Moments	Rel:None	Rel:Trust	Rel:Link	Rel:Cocitation
Mean	0.0000	-0.1481	0.0000	0.0000
Variance	0.8763	0.4068	0.4471	0.4059

Table 6.4: Statistics of Reviewer Quality Difference to Support Social Network Consistency Hypotheses.

three distributions are all symmetric with mean zero. However, Rel:Cocitation and Rel:Link have a much more concentrated peak around zero, i.e., smaller variance, compared with Rel:None. This supports the Co-Citation and Link Consistency Hypotheses that reviewers are more similar in quality (quality difference closer to zero) if they are co-trusted by others, or linked in a trust graph regardless of direction.

In the results of the KS-test, we have only one high p-value, for Rel:Link and Rel:Cocitation, while all the other pairs have p-values close to zero. This implies that Rel:Trust, Rel:Cocitation, or Rel:Link do not come from the same distribution as Rel:None. This observation directly connects the quality of reviewers with their relations in the social network. The correlation

between Rel:Link and Rel:Cocitation could potentially be explained by the relatively high reciprocity ratio (the percentage of links in the Trust social network that are reciprocal), and the relatively high clustering coefficient [65] which measures the tendency of triples to form triangles.

In summary, our experiments indicate that there exists correlation between review quality, reviewer quality, and social context. For all the three data sets considered, the statistics support our hypotheses for designing the regularizers.

6.5.3 Prediction Performance

For all three datasets (Cellphones, Beauty, and Digital Cameras), we randomly split the data into training and testing sets: 50% of the products for training (R_{train}), and 50% for testing (R_{test}). We keep the test data fixed, while sub-sampling from the training data to generate training sets of different sizes (10%, 25%, 50% or 100% of the training data). Our goal is to study the effect of different amount of training data on the prediction performance. We draw 10 independent random splits, and we report test set mean and standard deviation for our evaluation metrics. A polynomial kernel is used to enrich the feature representation for the linear model. We fix the parameter α of Linear Regression to the value that gives the best performance for the text-based baseline. Then, we report the best prediction performance by tuning the regularization weight β. We will discuss the parameter sensitivity in Section 6.5.3, while leaving the automatic optimization of parameters as future work.

We evaluate the effectiveness of different prediction methods using Mean Squared Error (MSE) over the test set R_{test} of size n_t,

$$MSE(R_{\text{test}}) = \frac{1}{n_t} \sum_{i=1}^{n_t} \left(Q(\mathbf{r}_i) - q_i\right)^2$$

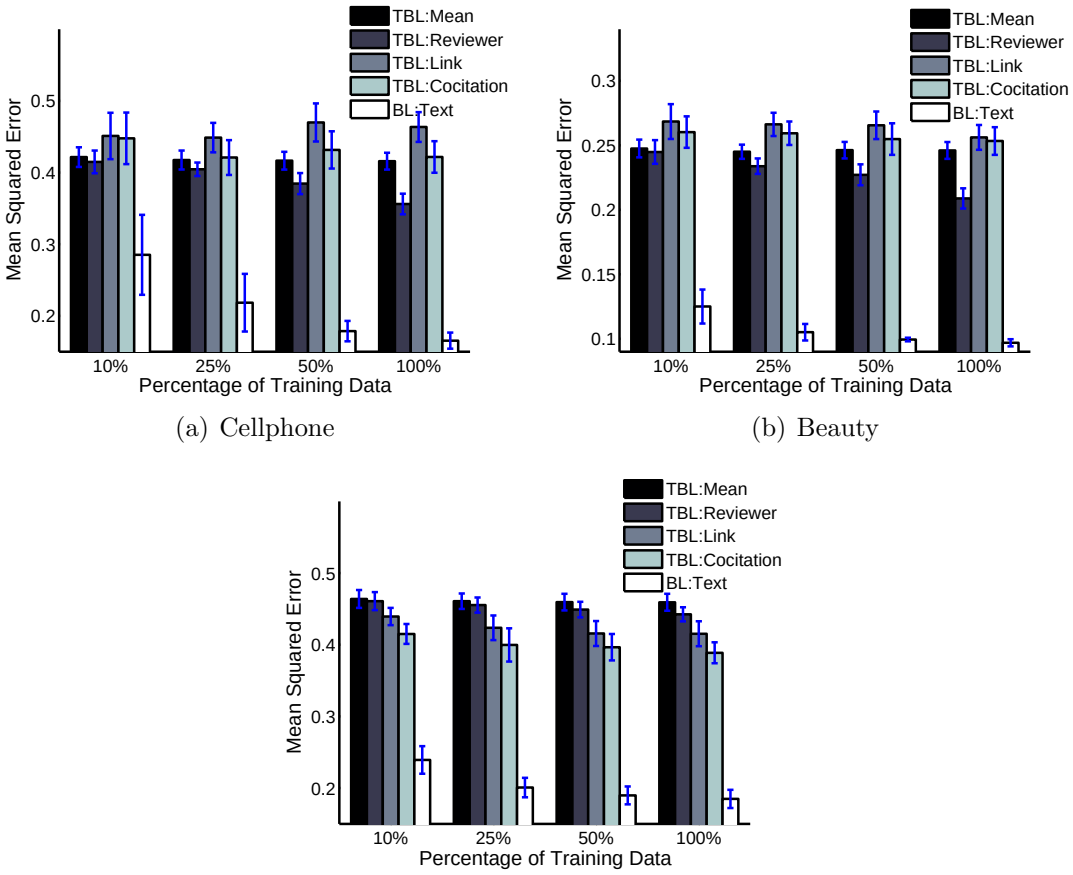
MSE measures how much our predicted quality deviates from the true quality. A smaller value indicates a more accurate prediction.

Simple Text-free Baselines

Since the graph statistics in Section 6.5.2 support our design of regularizers, we will examine a few text-free baselines (TBL) that are based solely on social context. These baselines also serve as a sanity check for the experiments we report in the following section. For the following, r denotes a test review written by reviewer u_r, and $\bar{Q}^*(u)$ is the quality of reviewer u as defined in Equation 6.9, when computed over the training data. If reviewer u has no reviews in the training data, $\bar{Q}^*(u)$ is undefined. We consider the following baselines for predicting the quality of r.

- **TBL:Mean**: Simply predict as the mean review quality in the training data $R_{\textbf{train}}$, i.e., $Q(r) = \frac{1}{n_t}\sum_{i=1}^{n_t} q_i$.
- **TBL:Reviewer**: Predict as the quality $\bar{Q}^*(u_r)$ of the author u_r in the training data. If it is not defined, predict as TBL:Mean.
- **TBL:Link**: Predict as the mean quality of all the reviewers connected to u_r in the link graph; if no such reviewer exists in the training set, or the value is undefined simply predict as TBL:Mean.
- **TBL:CoCitation**: Similar to TBL:Link, predict as the mean quality of all reviewers connected to u_r in the Co-Citation graph. If this is not defined predict as TBL:Mean.

We compare the four simple text-free baselines against **BL:Text**: the Linear Regression baseline that uses only text information. Figure 6.4 shows the MSE with standard deviation where the x-axis corresponds to the different percentages of the training data we used. We observe that none of the text-free baselines works as well as Linear Regression with textual features, suggesting that social context by itself cannot accurately predict the quality of a review. The MSE of the text-free baselines is lower for the Beauty category, where quality distribution is highly skewed at 4, but the text-based predictor is still significantly better. Out of the three social-context based baselines, TBL:Reviewer appears to provide more



(a) Cellphone

(b) Beauty

(c) Digital Cameras

Figure 6.4: MSE of Simple Text-free Baselines V.S. Text-only Baseline.

accurate prediction than the other two when there is rich social context (Cellphones and Beauty), but it offers marginal improvements over TBL:Mean in the case where the social context is sparse (Digital Cameras). TBL:CoCitation consistently outperforms TBL:Link, which is in line with our observation in Table 6.4 that the variance of Rel:Cocitation is smaller than that of Rel:Link.

Incorporating Social Context

We now compare the different techniques for review quality prediction that make use of text and social context of reviews. We consider the following methods.

- **BL:Text**: Linear Regression described in Section 6.3 (Equation 6.2) using only textual

Training Subset	10%	25%	50%	100%
Cellphone				
BL:Text	0.2852±0.0558	0.2183±0.0402	0.1787±0.0143	0.1654±0.0112
BL:Text+Rvr	0.3137±0.1079(9.99%)	0.2249±0.0518(3.02%)	0.1728±0.0116(-3.30%)	0.1552±0.0095**(-6.17%)**
REG:Link	0.2642±0.0292(-7.36%)	0.2113±0.0294(-3.21%)	0.1781±0.014(-0.34%)	0.1652±0.0111(-0.12%)
REG:CoCitation	0.2635±0.0359(-7.61%)	0.2064±0.0226(-5.45%)	0.1771±0.0133(-0.90%)	0.1647±0.0107(-0.42%)
REG:Trust	0.2563±0.0317(-10.13%)	0.2035±0.0205(-6.78%)	0.1768±0.0134(-1.06%)	0.1647±0.0108(-0.42%)
REG:Reviewer	0.2468±0.0223**(-13.46%)**	0.1958±0.0116**(-10.31%)**	0.1728±0.01**(-3.30%)**	0.1635±0.0089(-1.15%)
Beauty				
BL:Text	0.125±0.0132	0.1051±0.0064	0.0994±0.0014	0.0969±0.0028
BL:Text+Rvr	0.122±0.0123(-2.40%)	0.0973±0.0062**(-7.42%)**	0.089±0.002**(-10.46%)**	0.0857±0.0027**(-11.56%)**
REG:Link	0.1174±0.0073(-6.08%)	0.1036±0.0054(-1.43%)	0.0991±0.0016(-0.30%)	0.0968±0.0028(-0.10%)
REG:CoCitation	0.1166±0.007(-6.72%)	0.1036±0.0054(-1.43%)	0.099±0.0016(-0.40%)	0.0968±0.003(-0.10%)
REG:Trust	0.1157±0.0058(-7.44%)	0.1022±0.0044(-2.76%)	0.0986±0.0021(-0.80%)	0.0966±0.0029(-0.31%)
REG:Reviewer	0.112±0.0063**(-10.40%)**	0.1021±0.0049(-2.85%)	0.0984±0.0018(-1.01%)	0.0964±0.0028(-0.52%)
Digital Camera				
BL:Text	0.2392±0.0192	0.2007±0.0136	0.1897±0.0125	0.1848±0.0127
BL:Text+Rvr	0.2541±0.0239(6.23%)	0.2011±0.0106(0.20%)	0.1869±0.0096**(-1.48%)**	0.1801±0.0115**(-2.54%)**
REG:Link	0.2355±0.0211(-1.55%)	0.2002±0.0125(-0.25%)	0.1894±0.0124(-0.16%)	0.1848±0.0127(0.00%)
REG:CoCitation	0.2346±0.0204(-1.92%)	0.1994±0.0132(-0.65%)	0.1893±0.0126(-0.21%)	0.1848±0.0126(0.00%)
REG:Trust	0.2302±0.0183**(-3.76%)**	0.1984±0.0127**(-1.15%)**	0.189±0.0124(-0.37%)	0.1846±0.0127(-0.11%)
REG:Reviewer	0.2373±0.0189(-0.79%)	0.2005±0.0135(-0.10%)	0.1896±0.0124(-0.05%)	0.1848±0.0127(0.00%)

Table 6.5: MSE of Using Social Context as Features and as Regularization vs. Text-based Baseline

features.

- **BL:Text+Rvr**: Linear Regression described in Section 6.4.1 using both textual, and social context features.
- **REG:Reviewer**: Linear Regression with a regularizer under Reviewer Consistency Hypothesis (Equation 6.4).
- **REG:Link**: Linear Regression with a regularizer under Link Consistency Hypothesis (Equation 6.8).
- **REG:Cocitation**: Linear Regression with a regularizer under Cociation Consistency Hypothesis (Equation 6.7).
- **REG:Trust**: Linear Regression with a regularizer under Trust Consistency Hypothesis (Equation 6.6)

It is possible to consider combinations of the different regularizers. This would introduce multiple β parameters (one for each regularizer), and careful tuning is required to make the technique work. We defer the exploration of this idea to future work.

The results of our experiments are summarized in Table 6.5 where we show the mean MSE and the standard deviation for all techniques, over all categories, for different training data sizes. In the parentheses we have the percentage of reduction over MSE of the text-based baseline BL:Text. The best result (largest decrease of MSE) for each data set and each training size is emphasized in bold.

The first observation is that adding social context as additional features **BL:Text+Rvr** can improve significantly over the text-only baseline when there is sufficient amount of training data. The more training data available, the better the performance. BL:Text+Rvr gives the best improvement for training percentage of 50% and 100% for all three categories. We expect a similar trend for larger amounts of training data. On the other hand, when there is little training data, the social context features are too sparse to be helpful, and it

may be the case that the MSE actually increases, e.g., when training with 10% and 25% of the training data for Cellphone, and training with 10% for Digital Cameras. There are techniques for dealing with sparse data, however, exploring such techniques is beyond the scope of this work.

Using social context as regularization (method names starting with **REG**) consistently improves over the text-only baseline. The advantage of the regularization methods is most significant when the training size is small, e.g. using training percentage of 10% and 25% in all three data sets. This is often the case in practice, where we have limited resources for obtaining labeled training data, while there are large amounts of unlabeled data available.

Among the different regularization techniques, for both Cellphone and Beauty reviews, where there is relatively rich social context information, **REG:Reviewer** appears to be the most effective. For the Cellphone dataset, REG:Reviewer outperforms **BL:Text+Rvr** even with 50% of training data, indicating that social context regularization can be helpful when we have rich social context and balanced data. Among the regularization methods using the social network, **REG:Trust**, which is based on the most reasonable hypothesis, performs best in practice. This means that the direction of the trust social network carries more useful information than the simplified undirected link graphs and co-citation graphs.

Finally, for the Digital Camera reviews where the social context is very sparse there is still some improvement observed using regularization when the training data is small, but the improvement is not as significant as on the other two categories where the social context is richer; that is exactly what we expected.

In addition to the experiments on our test data, we are interested in testing our algorithms on data for which we have no social context information. Our premise is that using regularization can help to incorporate signals from the social network to the text-based predictor, thus improving accuracy prediction even if social context is not available. We now validate this premise. We use the Cellphone dataset, and we consider the case where we train on 10% of the training data. Within the test data of Cellphone, there is a subset of

Test on	Size	REG:Link	REG:CoCitation	REG:Trust	REG:Reviewer
All	1066	7.36%	7.61%	10.13%	13.46%
Reviews with no social context	144	3.33%	1.08%	3.15%	6.63%
Reviews with social context	922	8.11%	8.84%	11.47%	14.75%
Held-out reviews with hidden social context	893	10.38%	9.64%	11.73%	11.34%

Table 6.6: Improvement of Regularization Methods over BL:Text (Cellphone)

data (144 reviews on average across splits) that has no social context information, i.e., the author has only one review, and is not in the social network.[2] Regularization methods only adjust weights on textual features and are thus applicable to those anonymous reviews too, even though these reviews do not contribute to the added regularization terms. In Table 6.6, we report the percentage of improvement of four regularization methods over BL:Text. We still observe some improvement on anonymous reviews with no social context, although as expected less than on reviews with social context. This indicates the the generalizablity of the proposed regularization methods.

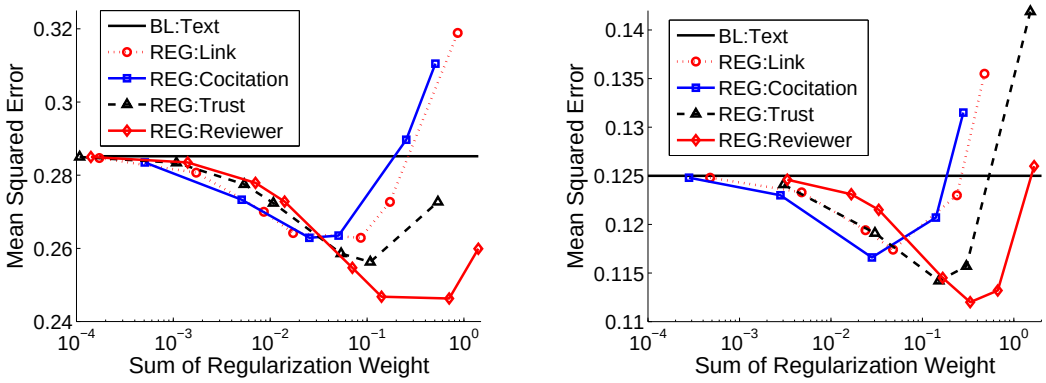
To further support the generalizablity claim, we try an extra set of experiments testing our regularization methods on a held-out set of reviews which are not used in the optmization process and for which we use only the textual features and hide their social context. More specifically, after learning a quality prediction function Q using 10% of the training data, we apply it to the remaining 90% of the training data, by multiplying the learned weight vector $\mathbf{w}$ with the text feature vectors of the held-out reviews. From the last row in Table 6.6, we can clearly see that compared with the text-only baseline, all regularization methods can learn a better weight vector $\mathbf{w}$ that captures more accurately the importance of textual features for predicting the true quality on the held-out set.

In summary, we make the following observations.

[2]Although we prune the data by requiring that each reviewer has at least two reviews and a link in the social network, due to multiple consecutive pruning conditions some reviewers end up with only one review and no links in the final pruned subset.

- Adding social context as features is effective only when there is enough training data to learn the importance of those additional features.
- On the other hand, regularization methods work best when there is little training data (which is a prevalent situation especially when we want to aggregate opinions from different sources) by exploiting the constraints defined by the social context and the large amount of unlabeled data (which is freely available online).
- Since regularization techniques incorporate the social context information into the text-based predictor, they provide improvements even when applied to data without any social context.

Parameter Sensitivity



(a) Cellphone (b) Beauty

Figure 6.5: Parameter Sensitivity.

Regularization methods have one parameter β to set: the trade-off weight for the regularization term. The value of the regularization weight defines our confidence in the regularizer: a higher value results in a higher penalty when violating the corresponding regularization hypothesis. In the objective functions (Equations 6.4, 6.6, 6.7, and 6.8), the contribution from the regularization term depends on β as well as the number of non-zero edges in the regularization graph. We define the sum of regularization weight as $\sigma = \beta \sum_{ij} \mathbf{M}_{ij}$, where

$\mathbf{M}$ can be the co-author matrix $\mathbf{A}$, the directed trust matrix $\mathbf{S}$, the co-citation matrix $\mathbf{C}$, or the undirected link matrix $\mathbf{B}$.

Figure 6.5 shows how the prediction performance of regularization methods varies as we use different values of σ. We only show the parameter sensitivity for Cellphone and Beauty reviews where the social context is relatively rich. The training data size is fixed to be 10%. As we can see, even though Cellphone and Beauty reviews carry different characteristics, the curves follow a very similar trend: as long as we set $\sigma \leq 0.1$, all regularization methods achieve consistently better performance than the baseline. As σ goes to zero, the performance converges to the text-based baseline. In addition, the shape of the performance curve depends on the corresponding hypothesis. For example, the optimum σ for REG:Trust is larger than that of REG:Link and REG:Cociation. Also, even with a value of σ higher than the optimum, the error of the REG:Reviewer does not increase as quickly as for the other methods. These observations are in line with the previous observations that the history of the reviewer (REG:Reviewer) and the Trust graph (REG:Trust) provide a better signal than the Co-Citation graph, or the Link graph.

6.6 Conclusion and Future Work

We studied the problem of automatically determining review quality using social context information. We studied two methods for incorporating social context in the quality prediction: either as features, or as regularization constraints, based on a set of hypotheses that we validated experimentally. We have demonstrated that prediction accuracy of a text-based classifier can greatly improve, when working with little training data, by using regularization on social context. Importantly, our regularization techniques make the general approach applicable even when social context information is unavailable. The method we propose is quite generalizable and applicable for quality (or attribute) estimation of other types of user-generated content. This is a direction that we intend to explore further.

As further future work, social context can be enhanced with additional information about items and authors. Information about product attributes, for example, enables estimates of similarity between products, or categories of products which can be exploited as additional constraints. Furthermore, although a portal may lack an explicit trust network, we plan to construct an implicit network using the ratings reviewers attach to each others' reviews and then apply our techniques to this case. Finally, rather than predicting the quality of each review, it would be interesting to adapt our techniques for computing a ranking of a set of reviews.

Chapter 7

Conclusions

In this chapter, we summarize the contributions of this thesis and discuss some interesting future research directions.

7.1 Summary

This thesis studies the problem of opinion integration and summarization for the goal of helping users better understand all the opinions for an arbitrary topic. However, to extract useful semantics from opinions is not trivial, especially since we want to apply the automatic methods to arbitrary topics. Unsupervised methods usually rely on many hand crafted heuristics that are domain/topic dependent, while supervised methods always require sufficient number of hand labeled training data. Neither of the two kinds of methods can easily adapt to a new domain, because of the cost of obtaining handmade heuristics/training examples. In this thesis, we propose a novel kind of approach that alleviates suh heavy dependence of human supervision. Our idea is to exploit many resources that are naturally available across different domains, such as structured ontologies and social networks. Such resources inherently carry domain specific information, thus provide helpful guidance. On the one hand, it is similar to unsupervised methods, because we do not need direct labels for the target semantics we want to infer, e.g., sentiment polarity of a word/phrase/aspect. On the other hand, it is similar to supervised methods, because we do acquire useful constraints from these resources, e.g., the combined sentiment of all the words/phrases/aspects should not deviate much from the observed overall sentiment rating. In this way, we offer a

general and robust line of methods applicable to multiple domains without requiring human supervision.

Along this general idea, we have done work in the following synergistic directions toward the goal of automatically generating integrated opinion summary: (1) exploiting well-structured resources (i.e., overview articles and structured ontology) to integrating relevant opinions from all kinds of Web 2.0 sources and summarize them along different aspects of the topic; (2) exploiting general-purpose sentiment lexicon, thesaurus of synonyms/antonyms, and overall sentiment ratings associated with opinion text, for inferring the sentiments in the opinions with respect to different aspects; (3) exploiting social interactions for predicting user level sentiment; and (4) exploiting social context (author information and social networks) for improving quality prediction of user generated opinions. Experimental results show that our proposed methods are effective and general enough to be applied for potentially many interesting applications in multiple domains such as business intelligence and political science.

7.2 Future Work

We have introduced a new problem of opinion integration and summarization, which aims at assisting users for easier digestion of large amounts of opinions. We have proposed robust and effective methods to automatically extract semantics (e.g., aspects, sentiments, and quality) out of the opinions which serve as essential components in the integrated summary. However, these are only the initial steps toward building a useful practical system. There are still many interesting directions of future work, including:

Topic-Relevant Opinion Retrieval: So far, we have assumed that the opinions relevant to a given topic are collected as a preprocessed step using information retrieval techniques. For example, we use Google to search for all blog articles mentioning "Barack Obama". However, although Google does a great job in returning a few most relevant results at the

top, many of the lower ranked results are not as accurate. Furthermore, even if one article is relevant to "Barack Obama", not every sentence is on this topic (e.g., one paragraph may be dedicated to discussing his opponent). Thus, it is interesting future work to study more advanced techniques to identify the truly relevant opinion sentences so that the following steps (as the methods we introduced in this thesis) can generate a more accurate integrated opinion summary.

Toward Personalization: The ideal process of searching for public opinion is essentially different depending on the user involved. Thus, it is important to make the opinion integration process adaptive to different user needs, i.e., produce personalized opinion integration and summary. In Section 3.1, we have already proposed and explored a principled way of incorporating user specified keywords for different aspects into a probabilistic integration model as conjugate priors. It would be interesting to further study how to personalize opinion integration without requiring a user to specify keywords. For example, we can model user interests implicitly based on the user past query history, click-through information, etc.

Toward Large-scale: While most information retrieval techniques have been optimized for efficiency, many useful text mining techniques, such as topic models, are still computationally expensive and difficult to be applied in large-scale or real-time tasks (which has also been revealed in our recent study outside this thesis [85]). It is a high impact future direction to develop more scalable techniques that can handle and analyze large amounts of text data quickly enough to support real-time interactive analysis. In particular, we believe that advancing the current text mining techniques with emerging technologies such as parallel computing and cloud-based infrastructures is a promising direction which can foster many exciting real-time text mining applications.

Toward Comparative Summary: When searching for people's opinions, the user usually also have another information need that is to compare two topics, e.g., Barack Obama v.s. Hilary Cliton or iPhone v.s. Blackberry. A very interesting future direction is to extend our

current work to comparative opinion summarization, which will lead to useful applications. New challenges need to be tackled for this extension. For example, when comparing two topics, it is more useful to identify the aspects that they are mostly compared at, and then organize opinions based on these "comparable" aspects.

To summarize, this thesis introduced a new problem of opinion integration and summarization. We believe that by exploiting naturally available resources, there are numerous opportunities in making the integrated opinion summary more useful and accurate. We anticipate that a more intelligent system can integrate all the approaches together and eventually change the way users search and understand opinions.

References

[1] *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*. ACM, 2007.

[2] Brevard user's group - technical glossary. *Brevard User's Group*, 2008-04-28.

[3] Glossary of technical terms. *Green Web Design*, 2008-04-28.

[4] Internet forum, 2011. `http://en.wikipedia.org/wiki/Internet_forum#cite_note-4`.

[5] Jacob Abernethy, Olivier Chapelle, and Carlos Castillo. Web spam identification through content and hyperlinks. In *AIRWeb '08*, pages 41–44, New York, NY, USA, 2008. ACM.

[6] Eugene Agichtein, Carlos Castillo, Debora Donato, Aristides Gionis, and Gilad Mishne. Finding high-quality content in social media. In *WSDM*, pages 183–194. ACM, 2008.

[7] Rakesh Agrawal, Sridhar Rajagopalan, Ramakrishnan Srikant, and Yirong Xu. Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th international conference on World Wide Web*, WWW '03, pages 529–535. ACM, 2003.

[8] Bazaarvoice. Social commerce statistics. In *http://www.bazaarvoice.com/resources/stats*.

[9] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

[10] Jiang Bian, Yandong Liu, Ding Zhou, Eugene Agichtein, and Hongyuan Zha. Learning to recognize reliable users and content in social media with coupled mutual reinforcement. In Juan Quemada, Gonzalo León, Yoëlle S. Maarek, and Wolfgang Nejdl, editors, *WWW*, pages 51–60. ACM, 2009.

[11] Giuseppe Carenini, Raymond T. Ng, and Ed Zwart. Extracting knowledge from evaluative text. In *K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture*, pages 11–18, New York, NY, USA, 2005. ACM.

[12] Kam Tong Chan and Irwin King. Let's tango — finding the right couple for feature-opinion association in sentiment analysis. In *PAKDD '09*, pages 741–748, 2009.

[13] Hao Chen and Susan Dumais. Bringing order to the web: automatically categorizing search results. In *CHI '00: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152, New York, NY, USA, 2000. ACM.

[14] Yejin Choi and Claire Cardie. Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *EMNLP '09*, pages 590–598, 2009.

[15] comScore/the Kelsey group. Online consumer-generated reviews have significant impact on offline purchase behavior. In *http://www.comscore.com/press/ release.asp?press=1928*, 2007.

[16] Hang Cui, Vibhu Mittal, and Mayur Datar. Comparative experiments on sentiment classification for online product reviews. In *Twenty-First National Conference on Artificial Intelligence*, 2006.

[17] Cristian Danescu-Niculescu-Mizil, Gueorgi Kossinets, Jon Kleinberg, and Lillian Lee. How opinions are received by online communities: a case study on amazon.com helpfulness votes. In *WWW '09*, pages 141–150, New York, NY, USA, 2009. ACM.

[18] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03*, pages 519–528, 2003.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statist. Soc. B*, 39:1–38, 1977.

[20] A. Devitt and K. Ahmad. Sentiment polarity identification in financial news: A cohesion-based approach. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 984–991, 2007.

[21] Xiaowen Ding, Bing Liu, and Philip S. Yu. A holistic lexicon-based approach to opinion mining. In *WSDM '08*, pages 231–240.

[22] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[23] Michel Galley, Kathleen McKeown, Julia Hirschberg, and Elizabeth Shriberg. Identifying agreement and disagreement in conversational speech: use of bayesian networks to model pragmatic dependencies. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL '04. Association for Computational Linguistics, 2004.

[24] Michael Gamon, Anthony Aue, Simon Corston-Oliver, and Eric K. Ringger. Pulse: Mining customer opinions from free text. In A. Fazel Famili, Joost N. Kok, José María Peña, Arno Siebes, and A. J. Feelders, editors, *IDA*, volume 3646 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2005.

[25] Anindya Ghose and Panagiotis G. Ipeirotis. Designing novel review ranking systems: predicting the usefulness and impact of reviews. In *ICEC '07*, pages 303–310, New York, NY, USA, 2007. ACM.

[26] Erving Goffman. *Frame Analysis: An essay on the organization of experience*. Cambridge: Harvard University Press, 1974.

[27] A.B. Goldberg and X. Zhu. Seeing stars when there arent many stars: Graph-based semi-supervised learning for sentiment categorization. In *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing*, 2006.

[28] Ahmed Hassan and Dragomir Radev. Identifying text polarity using random walks. In *ACL '10*, pages 395–403, Morristown, NJ, USA, 2010. Association for Computational Linguistics.

[29] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics, 1997.

[30] Peter Hitchens. The broken compass: How british politics lost its way. In *Continuum International Publishing Group Ltd. ISBN 1847064051.*, 2009.

[31] Linh Hoang, Jung-Tae Lee, Young-In Song, and Hae-Chang Rim. Combining local and global resources for constructing an error-minimized opinion word dictionary. In *PRICAI '08*, pages 688–697, Berlin, Heidelberg, 2008. Springer-Verlag.

[32] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[33] John A. Horrigan. Online shopping. In *Pew Internet & American Life Project Report*, 2008.

[34] M. Hu and B. Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM New York, NY, USA, 2004.

[35] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.

[36] Minqing Hu and Bing Liu. Mining opinion features in customer reviews. In Deborah L. McGuinness and George Ferguson, editors, *AAAI*, pages 755–760. AAAI Press / The MIT Press, 2004.

[37] Valentin Jijkoun, Maarten de Rijke, and Wouter Weerkamp. Generating focused topic-specific sentiment lexicons. In *ACL '10*, pages 585–594. Association for Computational Linguistics, 2010.

[38] Yohan Jo and Alice Oh. Aspect and sentiment unification model for online review analysis. In *WSDM '11*.

[39] Mika Käki. Optimizing the number of search result categories. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*, pages 1517–1520, New York, NY, USA, 2005. ACM.

[40] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *EMNLP '06*, pages 355–363, Morristown, NJ, USA, 2006. Association for Computational Linguistics.

[41] V. O. Key. *Public Opinion and American Democarcy*. Publisher: John Wiley, New York, 1964.

[42] Soo-Min Kim and Eduard Hovy. Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367, 2004.

[43] Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. Automatically assessing review helpfulness. In *EMNLP*, pages 423–430, Sydney, Australia, July 2006.

[44] Peter Kollock and Marc Smith. *Communities in Cyberspace*. Routledge Press, 2001.

[45] Anton V. Leouski and W. Bruce Croft. An evaluation of techniques for clustering search results. Technical report, 1996.

[46] Kevin Lerman, Sasha Blair-Goldensohn, and Ryan T. McDonald. Sentiment summarization: Evaluating and learning user preferences. In *EACL*, pages 514–522. The Association for Computer Linguistics, 2009.

[47] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA, 2005. ACM.

[48] Jingjing Liu, Yunbo Cao, Chin-Yew Lin, Yalou Huang, and Ming Zhou. Low-quality product review detection in opinion summarization. In *EMNLP-CoNLL*, pages 334–342, 2007. Poster paper.

[49] Yang Liu, Xiangji Huang, Aijun An, and Xiaohui Yu. Modeling and predicting the helpfulness of online reviews. In *ICDM*, pages 443–452. IEEE Computer Society, 2008.

[50] L. Lovasz and M. Plummer. Matching theory. In *Annals of Discrete Mathematics*, North Holland, Amsterdam, 1986.

[51] Yue Lu and Chengxiang Zhai. Opinion integration through semi-supervised topic modeling. In *WWW '08*, pages 121–130, New York, NY, USA, 2008. ACM.

[52] Yue Lu, ChengXiang Zhai, and Neel Sundaresan. Rated aspect summarization of short comments. In *WWW '2009*, April 2009.

[53] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[54] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 101–110, New York, NY, USA, 2008. ACM.

[55] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW '07*, pages 171–180. ACM.

[56] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW* [1], pages 171–180.

[57] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW* [1], pages 171–180.

[58] Qiaozhu Mei, Chao Liu, Hang Su, and ChengXiang Zhai. A probabilistic approach to spatiotemporal theme pattern mining on weblogs. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 533–542, 2006.

[59] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In Pavel Berkhin, Rich Caruana, and Xindong Wu, editors, *KDD*, pages 490–499. ACM, 2007.

[60] Qiaozhu Mei and ChengXiang Zhai. A mixture model for contextual text mining. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 649–655, 2006.

[61] Saif Mohammad, Cody Dunne, and Bonnie Dorr. Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. In *EMNLP '09*, pages 599–608. Association for Computational Linguistics, 2009.

[62] Akiko Murakami and Rudy Raymond. Support or oppose?: classifying positions in online debates from reply activities and opinion expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 869–875, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[63] Seung-Hoon Na, Yeha Lee, Sang-Hyob Nam, and Jong-Hyeok Lee. Improving opinion retrieval based on query-specific sentiment lexicon. In *ECIR '09*, pages 734–738, Berlin, Heidelberg, 2009. Springer-Verlag.

[64] A. Neviarouskaya, H. Prendinger, and M. Ishizuka. Sentiful: Generating a reliable lexicon for sentiment analysis. In *ACII*, pages 1 –6, sep. 2009.

[65] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[66] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*, pages 115–124, 2005.

[67] Bo Pang and Lillian Lee. *Opinion Mining and Sentiment Analysis*, volume 2(1–2) of *Foundations and Trends in Information Retrieval*. Now Publ., 2008.

[68] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *EMNLP '02*, pages 79–86, 2002.

[69] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86, 2002.

[70] Ana-Maria Popescu and Oren Etzioni. Extracting product features and opinions from reviews. In *HLT '05*, pages 339–346, Morristown, NJ, USA, 2005. Association for Computational Linguistics.

[71] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[72] Lee Rainie and John Horrigan. Election 2006 online. In *Pew Internet & American Life Project Report*, 2007.

[73] Christina Sauper and Regina Barzilay. Automatically generating wikipedia articles: A structure-aware approach. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 208–216, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[74] B. Snyder and R. Barzilay. Multiple aspect ranking using the good grief algorithm. In *Proceedings of NAACL HLT*, pages 300–307, 2007.

[75] Tao Tao and ChengXiang Zhai. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 162–169, 2006.

[76] Matt Thomas, Bo Pang, and Lillian Lee. Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of EMNLP*, pages 327–335, 2006.

[77] I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL '08*, pages 308–316.

[78] Ivan Titov and Ryan McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, June 2008.

[79] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 111–120, New York, NY, USA, 2008. ACM.

[80] Oren Tsur and Ari Rappoport. Revrank: a fully unsupervised algorithm for selecting the most helpful book reviews. In *ICWSM*, 2009.

[81] Peter D. Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 417–424, 2002.

[82] Peter D. Turney and Michael L. Littman. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst.*, 21(4):315–346, 2003.

[83] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *KDD '10*, pages 783–792, New York, NY, USA, 2010. ACM.

[84] Yue Lu, Huizhong Duan, Hongning Wang, and ChengXiang Zhai. Exploiting structured ontology to organize scattered online opinions. In *COLING '10: Proceedings of the 23rd International Conference on Computational Linguistics*, pages 734–742.

[85] Yue Lu, Qiaozhu Mei, and ChengXiang Zhai. Investigating task performance of probabilistic topic models - an empirical study of plsa and lda. *Information Retrieval*, 2010.

[86] Yue Lu, Panayiotis Tsaparas, Alexandros Ntoulas, and Livia Polanyi. Exploiting social context for review quality prediction. In *WWW '10: Proceeding of the 19th international conference on World Wide Web*, pages 691–700.

[87] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma, and Jinwen Ma. Learning to cluster web search results. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 210–217, New York, NY, USA, 2004. ACM.

[88] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of CIKM 2001*, pages 403–410, 2001.

[89] ChengXiang Zhai, Atulya Velivelli, and Bei Yu. A cross-collection mixture model for comparative text mining. In *Proceedings of KDD '04*, pages 743–748, 2004.

[90] Tong Zhang, Alexandrin Popescul, and Byron Dom. Linear prediction models with graph regularization for web-page categorization. In *KDD*, pages 821–826. ACM, 2006.

[91] Zhu Zhang and Balaji Varadarajan. Utility scoring of product reviews. In *CIKM '06*, pages 51–57, New York, NY, USA, 2006. ACM.

[92] Jing Zhao and Jing He. Learning to generate labels for organizing search results from a domain-specified corpus. In *WI '06: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 390–396, Washington, DC, USA, 2006. IEEE Computer Society.

[93] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *NIPS*. MIT Press, 2003.

[94] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[95] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919. AAAI Press, 2003.

[96] Xiaojin Zhu and Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.