REVIEWMINER: A NOVEL SYSTEM FOR MULTI-MODAL REVIEW ANALYSIS TO PROVIDE VISUALIZED SUPPORT FOR DECISION MAKING

BY

GONG CHEN

THESIS
Submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Adviser:
Professor Chengxiang Zhai
Abstract

Over the past decades, a number of prominent sites for collecting and presenting reviews in different product categories have seen tremendous growth. However, the growth of number and length of the review text doesn’t provide easier access to knowledge in the review. In fact as the amount of reviews grow, people are less likely to be able to finish reading the helpful reviews. To tackle the information overload situation in review text data, I developed features on top of an existing review analyzer named ReviewMiner. It is a novel system for understanding review data. ReviewMiner is a multi-modal system that combines visualizations of spatial, temporal as well as textual summary of aspects of the review. The highly rich content produced from large amounts of review data empowers the user of the system to make more informed decisions. Because of the underlying LARA algorithm for general aspect rating analysis, ReviewMiner is a framework that can be applied to many categories of review data such as hotel review, restaurant reviews, product review, medical review, and more. In the focused study of hotel review scenario, I demonstrate the system’s major components: (1) natural language search; (2) visualizing the review data with location attributes; (3) analysis of temporal evolution of aspect rating; (4) latent aspect analysis on review text data.
Acknowledgments

This thesis would not be possible without the excellent support from a group of people who have helped me in development of the system and insightful guidance in the ideation.

First of all, I would like to thank my advisor, Professor Chengxiang Zhai, for his generous encouragement, patience, and more importantly, his trust and respect which has motivated me to drive for quality results. He has guided me on identifying interesting research problems and has given me the freedom to explore many possible solutions. By always being supportive and helpful, Cheng has been a driving component of my academic career.

The work relies on excellent original research of Hongning Wang’s LARA algorithm and previous development of his ReviewMiner system. As a senior member of Professor Zhai’s group, he has been a bottomless source of encouragement, understanding, and true inspiration. Hongning’s continuous support and invaluable suggestions has made my thesis development not only painless but also of superior quality.

I want to also thank Jinyao Xu for being a great friend and co-developer. The stressful nights we spent together developing the system and fixing bugs is surely going to be some of the most memorable moments of my university years.
Table of Contents

1. Introduction ...................................................................................................................... 1
   1.1 Existing System and Research .................................................................................. 2
   1.2 ReviewMiner System ............................................................................................... 3
   1.3 Organization of the Thesis ....................................................................................... 4
2. Related Work .................................................................................................................... 5
3. System Architecture ........................................................................................................ 6
   3.1 Backend Dataflow ..................................................................................................... 6
   3.2 Frontend Pages ......................................................................................................... 7
4. Natural Language Query .................................................................................................. 9
   4.1 Motivation ................................................................................................................ 10
   4.2 Implementation ......................................................................................................... 11
5. Location Attribute Visualization .................................................................................... 12
   5.1 Motivation ................................................................................................................ 12
   5.2 Implementation ......................................................................................................... 14
6. Temporal Evolution of Ratings ...................................................................................... 15
   6.1 Motivation ................................................................................................................ 16
   6.2 Implementation ......................................................................................................... 16
7. Opinion Mining in Aspects ............................................................................................. 17
   7.1 Motivation ................................................................................................................ 18
   7.2 Implementation ......................................................................................................... 18
8. Conclusion and Future Work .......................................................................................... 19
References ........................................................................................................................... 21
1. Introduction

In this era of social shopping, more and more people start taking advices from other users’ experiences reported online in the form of reviews. There are many prominent sites for different domain of reviews. Amazon, being one of the largest online retailer marketplaces, contains an incredible amount of review information. Their product categories span from small items such as electronics, books, and clothing to home improvement, auto parts [1]. Many of these products have received a great number of reviews. Other sites including Yelp for local businesses reviews (mainly restaurants) and TripAdvisor for hotel reviews and ratings have accumulated millions of reviews about their specific product category [2] [3]. The growth of online reviews presents a great opportunity for information retrieval. The human curated reviews contain vast amount of valuable knowledge about certain entities. From reading the reviews, someone who has never been to a particular hotel can obtain a “word of mouth” recommendation on whether the hotel is worth staying or not. The information contained in the reviews can be useful to making decisions such as ordering certain dish at a restaurant, picking out an ideal location to stay when visiting a foreign city, or comparing between several electronic products. However, digesting the enormous amount of review text data is painfully difficult for a normal user. When there are many reviews available on an entity, the user is not able to completely read every review. The unstructured nature of the text review data also causes ineffective aggregation of information for an average
review reader. The overloaded review text information presents a challenge to provide a detailed overview of text review information.

1.1 Existing System and Research

This challenge has inspired a myriad of research about how to accurately and efficiently gather knowledge from the large amount of review text data. Previous research has shown that it is quite possible to extract information from reviews [4] [5] [6], analyze the sentiment of reviews [7] [8] [9], and create a summary of the opinions contained in the reviews [10] [11] [12]. In many recent advances on opinion analysis [13] [14] [15], the proposed solutions are able to produce good quality and detailed opinions summaries about the reviews. ReviewMiner takes a step further to incorporate the visual information of location maps and review rating over time to better summarize the

Figure 1 shows the map view page of tripadvisor.com. Note the individual hotels have only the review counts and overall rating but the map does not show the comparison between hotel ratings.
information to provide support for more informed decision making.

1.2 ReviewMiner System

I have developed several key features and integrated with the current ReviewMiner system to address these problems. The ReviewMiner system is a novel system for multi-modal review analysis to provide visualized support for decision making. The new ReviewMiner system is now a general framework to incorporate rich graphical and textual information regarding the spatial, temporal, and the latent aspects of review data. Empowered by a new algorithm that automatically finds given number of aspects within the review [16], ReviewMiner is able to realize many applications. Demonstration results in the hotel domain show that compared with existing hotel search system, ReviewMiner can provide more useful functions such as showing the location heatmap of top rated hotels compared to surrounding hotels; producing temporal aspect ratings to support decision making in business intelligence; discovering aspects of the reviews that are most important to a normal hotel shopper; being able search in natural language query that put emphasis on interested aspects [16]. This multi dimensional visual presentation of text information enables a high level approach to quickly digest large amounts of reviews. ReviewMiner is useful to users who are interested in searching for the best hotel in his preferred aspects because by using a natural language query “Find Cheap Hotels in Chicago” the users can immediately see a distribution on the map showing the top rated hotels with his preference. This liberates the users from
reading the detailed reviews and creates a high level overview to help them make choices. The ReviewMiner system also provides a new way to help business professionals who read reviews to gather feedbacks about a product. Traditionally merchants selling products on the web often have to read through the customer reviews or summaries of customer reviews in order to gather feedback information to further improve their product [12]. With ReviewMiner, the business professionals can immediately view and compare timelines of changes in reviews’ rating, aspect mention of products of interest.

1.3 Organization of the Thesis

In order to fully explain the ReviewMiner system, this thesis is organized to the style of a demo paper. In Chapter 2 related work, this thesis briefly survey the existing research and systems and gives a context and inspiration for ReviewMiner. In Chapter 3 System Architecture, this thesis explains the design and main components of the system. In the chapters following Chapter 2, this thesis explains the context, implementation and usefulness for each major feature. These major features include visualizing the review data with location attributes; natural language search; showing aspect ratings’ evolution over time; latent aspect analysis on review text data. Chapter 8 concludes the work and the future possibilities and extensions can be made to improve the current system.
2. Related Work

The challenge presented by the information overload of large amount of unstructured text has inspired a myriad of research about how to accurately and efficiently gather knowledge. There are many works regarding opinion analysis on text information. At the basic level, past research has demonstrated classification of text information into positive and negative binary sentiments [7] [9]. This idea further developed into classifying into multi segment scale [8]. Previous attempts at addressing opinion analysis problem use a number of supervised un-supervised approaches. However, the resulting analyses fall short to give detailed aspect based rating. Instead, the prior work focuses on an overall rating. Previous research has also shown that it is quite possible to extract information from reviews [4] [5] [6], analyze the sentiment of reviews [7] [8] [9], and create a summary of the opinions contained in the reviews [10] [11] [12]. In many recent advances on opinion analysis [13] [14] [15], the proposed solutions are able to produce good quality and detailed opinions summaries about the reviews. There are also independent works on incorporating either the spatial or temporal dimension of review data [12] [17] [15]. However, to my best knowledge, there are no current system that can offer a rich and complete overview of review text including the location distribution and temporal evolution of the rating.
3. System Architecture

The system sports a Java based web application with struts for handling dynamic URL bindings. The system has two main components: the front end and the back end. The backend consists of a database, an algorithm analyzer module, a request handler for the frontend, and a database connector. The frontend consists of three pages.

3.1 Backend Dataflow

The backend dataflow is a straightforward one as seen in many web based applications. When a request for list of products comes in through the frontend, the frontend handler starts a search for the query terms into the database. The database is a central storage for all the review and business attribute information. Through the database connector, the algorithm analyzer module expose the data in the storage to calculate the aspects and rating scores used in ranking the list of products as the result. The frontend handler then publishes this result to the incoming request.

Figure 2 shows the backend processes and dataflow
3.2 Frontend Pages

The frontend is designed to be simple and elegant with minimalistic controls and settings to avoid confusions. When users first visit the site, they are presented with a search input box that provides auto-complete functionalities to the natural language queries. After the search, the users are presented with a list of top rated products on which the users can click through further to see detailed review and aspect information about the product of interest.

![Diagram showing workflow of pages](image)

*Figure 3 shows a workflow pages user see when they visit the ReviewMiner website*

At the search page user is able to type in natural language queries such as “Cheap hotels in Chicago” to retrieve a list of relevant hotels immediately. They can also login to their previous session to retrieve the user’s saved preferences for each aspect. At the product list page, the user can view a list of search results and a map showing a heatmap of highly rated areas. At this page, the user can also modify the preference option to produces a different ranking of the products. When the user further clicks through to
the product detail page, the user can view a temporal record of review rating changes per aspect, summarized review highlights as well as individual review texts.
4. Natural Language Query

Natural language processing is a well studied topic. Studies have shown that improved natural language processing quality also improves quality of the information retrieved [16] [17] [18]. In ReviewMiner natural language processing techniques help transforming the unstructured query into concrete specification objects which contain weighting information about different aspects of the product. We leverage this information to improve the search results. In this chapter I explain the front end work I have done to make the query processing task easier.

Figure 4 is a screen shot of the first page a user encounter when visiting ReviewMiner. This figure also shows the auto-complete helper for several example queries.
4.1 Motivation

Natural language query is a convenient choice of communication. It is the most intuitive way for a user to search for a specific product. However, modern systems cannot expect a casual user to input specific queries such as the exact addresses of interested hotels or the names of the restaurant to go to for dinner because the average user simply have no way to make a more intelligent decision other than random guessing. So the users often describe a rough idea of the desired features such as cheap price, good location. It is the system’s job to make sense of these rough ideas. Having the mapping from natural language to aspect specific requirements and weights can vastly improve the search experience and simplify the casual user’s thinking process in getting high quality results. The importance of this particular feature is to empower the casual users with effortless access to information.

In the demo hotel search scenario, I created an auto-complete feature at the search input. As the user types in the rough concept to search for, the search field interface automatically matches the text with existing aspect keywords such as cheap hotels, good service, and premium location. These auto-complete queries help the user input the query in a uniform way to avoid inaccuracies in processing. The system then interprets the natural language into aspect weights and conduct filtering and ranking on the gathered specifications.
4.2 Implementation

The implementation of the auto-complete uses JavaScript to actively match the input with regular expression search in an existing keyword mapping. The JavaScript matching relies on an open source twitter typeahead engine [19]. The matched string is then sent to the frontend handler for processing. After processing, the query is converted into specification objects that contain aspect weights mapped from the sentiment terms such as cheap price and good location. Each specification object stores all the information about one aspect. For example, a “find hotels with nice non-smoking rooms” can have a specification object about “room” aspect. The specification object contains the weights for “room” aspect, the default minimum rating for the “room” aspect, and the specific requirement of “non-smoking.” In the backend, this aspect weighting information in the specification object is linearly combined with user’s saved preference weights to be sent to the analyzer for calculation of ranking scores. Since queries can contain specific filter phrases such as “price less than $200” and “non-smoking room”. These are processed into filters in the specification objects. The filters are applied on the ranking of results to further improve the retrieval quality. Current demo only shows limited examples of auto-complete terms. The system can be easily extended to support concatenating terms separated by a predefined word such as “and,” “or,” or “,” within auto-complete vocabularies.
5. Location Attribute Visualization

Another new feature that I have developed and integrated with ReviewMiner system is the visualization of review’s location information. The location dimension visualizes the hotel review’s rating information. The visualization is accomplished with a heatmap that represent the distribution of high and low review rating of hotels in a city. The location feature simplifies the decision making process for a casual user. Digesting reviews based on the location information transforms the decision from choosing where to stay to which place has the higher rating near the target destination according to my preference.

5.1 Motivation

Location information is commonly used to show the exact address of the entity in review. It is a standard for review sites that deal with location sensitive elements such as hotel and restaurant information. Tripadvisor [2], for example, is a hotel booking and trip planning review site. It includes a map to show the locations for its hotels. Although the review information is accessible at each hotel’s page, in figure 1, the review information is not present on the map. The user must click on each hotel to gather the rating and review aspects. The workflow design has made comparing hotels based on user’s preference close to impossible because tripadvisor’s design requires the users to digest each hotel’s information first by clicking through and reading the reviews about
the current hotel before the users can compare with other hotels that have similar location.

I solve this problem by introducing an extra layer in the map to represent the review rating distribution at the location. With one glance at the map, the users efficiently obtain several pieces of information. At the same time user can see the top rated hotels marked on the map as well as the rating distribution surrounding the hotels. The combination of review information and the location information satisfy the need to compare hotels and effectively communicate the rating of certain areas. The users no longer need to dig deep into the hotel reviews before comparison, and their decision making process is simplified. The users can immediately see the areas that are red(with higher ratings) compared to lighter green(with lower ratings) and the general areas with the best rated hotel. With markers marking the top rated hotels in the area, users get quick suggestions of the hotels that match their preference.
5.2 Implementation

The second page of the frontend workflow displays the heatmap. When the search result returns, a small map with a heatmap layer is drawn. User can interact with the small map by dragging and zooming. When the user clicks on the map, a larger map appears to show more map area and marker detail. The heatmap consists of the data points of all the hotels in the search result. Then based on the hotel’s aspect rating, the data points can be assigned relevant weights to show relative rating differences. The map is using GoogleMap API [20].
6. Temporal Evolution of Ratings

Included in the original system, the temporal text mining is an interesting theme to study because it provides insight to the appearance of changing trends. There is existing research studying this broad topic [21]. ReviewMiner exposes a special use case of temporal feature on review data. It summarizes review aspects according to the time dimension to produce some interesting insight to how the product performs and which aspects are most important to users over a long period of time.

Figure 6 shows temporal view of the review text data. With one glance at the graph, the business professionals already know how the product developed over time and how the users received and cared about the product.
6.1 Motivation

For business professionals who read reviews to gather feedbacks about a product, ReviewMiner provides another useful modal of summary. Traditionally merchants selling products on the web often have to read through the customer reviews or summaries of customer reviews in order to gather feedback information to further improve their product [12]. However, the reading for each product is slow. Afterwards, the comparison requires a lot of human effort and is prone to subjective errors. With ReviewMiner, the business professionals can immediately view and compare timelines of changes in reviews’ rating, aspect mention of products of interest. For example, a product manager wishes to learn how a particular product is received over a period of time. With current systems, he has to go through each review to manually extract the relevant information. With ReviewMiner, he can quickly overview consumers’ focused aspects for a new product launch versus a seasoned product. Extracting knowledge of the evolution of a product can be valuable business intelligence to support product strategic decision making.

6.2 Implementation

The implementation for graphing relies on open source JavaScript libraries [22] [23]. The backend LARA algorithm used to extract aspect ratings over time is written in Java [24]. The implementation of this feature is based on the original ReviewMiner system.
7. Opinion Mining in Aspects

Opinion mining is extensively studied field with numerous works studying classification of sentiment on a given text corpus. However, these works aggregate the opinions into a general category instead of detailed aspects [12] [6] [5] [15]. Contained in the original ReviewMiner is a new algorithm that automatically discovers the latent aspects in the review text [24]. The aspects provide more relevant information to combat information overload large amounts of review text data.

Figure 7 shows the opinion aspects mined from review text data. Each aspect is color coded to the original sentences that discusses the detailed opinion about the aspect.

In Figure 7, I show the page showing the opinion mining feature within ReviewMiner. The aspect and the text corresponding to the aspect are color coded so that if a user is interested in learning the detailed comments on the aspect he can easily find the
location of the text. Each aspect has a summary section highlighting the positive and negative reviews regarding the aspect. Each aspect also contains a calculated score to represent the overall rating for this aspect of the product. In the hotel review scenario, users can browse the hotel’s Value, Room, Location, Cleanliness, and Service rating through a slide scroll at the center of the page. Without reading the details of the reviews, users can understand whether the hotel matches with their expectation.

7.1 Motivation

Information overload prevents people from getting useful knowledge from the large amount of review text. When given an overall rating or summary of the hotel, casual users have no way to identify whether the rating was given for excellent service, cheap price, or beautiful room and view. Without knowing the aspect information, users cannot know whether the hotel will meet their expectation. To answer these questions from users, ReviewMiner solves the problem by providing the aspect information to give context of the rating.

7.2 Implementation

The LARA algorithm is implemented to calculate aspect ratings for hotels. Then in the analyzer module in the backend, a ranking score is calculated based on the preference weights for each aspect and the aspect’s rating to produce an ordered list of hotels.
8. Conclusion and Future Work

This thesis describes a specific experiment with a general framework to visually present the review text data in a rich and informative overview covering the review text’s spatial, temporal, and aspect specific dimensions. This system is built to improve the review reading experience by providing high level graphical summaries to support decision making by both the casual user and the business professionals.

The ReviewMiner system contains many components some of which have not yet become a standard in the commercial space. The spatial representation of the review data is useful and a necessary component in the hotel review case study. The temporal changes of the review aspect ratings can be helpful to business professionals who need a simple way to aggregate review over time. The opinion analysis in different aspect provides a more contextual rating based on the specific aspect of the review. The aspect indication is a helpful guide to systematically digest the reviews in all the aspects for review readers.

One important contribution of this thesis is the extension and improvement of the system and its features. The ReviewMiner system focuses on providing an effortless experience to extract knowledge from review text data making it an ideal tool to simplify the lives of many people who have difficulty discerning the better choice.

ReviewMiner relies on a general algorithm to extract aspects and per aspect ratings. So a natural extension is to apply the system to extensive review data. With more review categories, the overview information can become richer and more credible. For example
when ReviewMiner contains reviews for schools, restaurants, hospitals etc, it is possible to provide concrete objective evidence to support for the hardest home buying decisions. Another improvement can be done to the input interface, an advanced interface for user to input specific requirements instead of solely relying on natural language processing can improve the accuracy of the queries.
References


