國立交通大學

資訊學院資訊科技（IT）產業研發碩士專班

碩士論文

Bookle：書籍推薦系統 基於讀者的評論與閱讀行爲

Bookle : A Book Recommender System Based on Readers' Reviews and Reading Behaviors

研究生：卓建益

指導教授：黃俊龍 教授

中華民國九十九年八月

Bookle ： 書籍推薦系統
基於讀者的評論與閱讀行爲

Bookle : A Book Recommender System
Based on Readers' Reviews and Reading Behaviors

研究生：卓建益 Student: Chien-I Tso
指導教授：黃俊龍 Advisor: Jiun-Long Huang

國立交通大學
資訊學院資訊科技（IT）產業研發碩士專班
碩士論文

A Thesis
Submitted to College of Computer Science
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
in
Industrial Technology R & D Master Program on
Computer Science and Engineering
August 2010
Hsinchu, Taiwan, Republic of China

中華民國九十九年八月

Bookle ： 書籍推薦系統
基於讀者的評論與閱讀行爲

學生：卓建益　　　　　　　　　　指導教授：黃俊龍

國立交通大學資訊學院資訊科技（IT）產業研發碩士專班

摘要

推薦系統就好比是一種獨特且專門的資訊搜尋引擎，在本篇論文中，我們嚐試從新的角度來開發書籍推薦系統，傳統的書籍推薦系統，大多透過記錄過去使用者的購買行爲來做推薦，或是要求使用者以填充特定資料的方式來過濾搜尋結果，而我們的做法是藉由使用者描述書籍內容的方式做爲搜尋依據，企圖找出更能貼近讀者需求的書籍。

在實作上，我們收集眾多的書籍評論作爲基礎資料，藉由資料探勘與資訊擷取技術，找出評論中的關鍵資訊，統整這些資訊後，進而推測出各書籍的主要內容，更進一步的，爲了評斷每篇評論的可信度與價值，系統利用閱讀行爲衡量評論品質，讓各篇評論獲得應有的價值，讓推薦的結果更具說服力。

Bookle : A Book Recommender System Based on Readers' Reviews and Reading Behaviors

Student: Chien-I Tso Advisor: Jiun-Long Huang

Industrial Technology R & D Master Program of
Computer Science College
National Chiao-Tung University

ABSTRACT

Recommender system is like a search engine for specific and specialized information. In this paper, we attempt to develop a book recommender system from a new perspective. Most of the traditional book recommender systems are developed through the records of purchasing behaviors of users, or specific words user are required to key in for filtering search results. However, what our system needs is the description about the book content that users are interested in. The purpose of our work is attempting to find the best books that meet the needs of readers.

In the implementation of the project, we collected a large amount of book reviews as source data. With data mining and information retrieval technology, we planed to find the critical information out from reviews and integrated the information, and then the main contents of books can be speculated. Furthermore, in order to determine the credibility and value of each review, the system would measure review quality by reading behaviors. So that the reviews would be correctly scored which makes the result of recommendation become more convincing.

誌　　　　謝

隨著九月的到來，我在交大碩班的日子漸告終結，在這段日子裡，我認識了很多新朋友，每個人都是我生命中的貴人。感謝我的指導老師黃俊龍老師引領我進入Web API的世界，讓我學習到很多新觀點與技術，開拓了我的視野，感謝在友訊研發中心認識了工作夥伴必安跟昱樺，我們在一起工作打混的有趣日子讓人難以忘懷，感謝研究室裡的好夥伴金亮、濬仲、信翰，跟你們在同一個實驗室裡學習成長，讓我感覺到我很幸運，你們每位都是好ㄉㄉ，我從你們身上學習了很多，感謝辛苦幫我修改論文的士詮學長，抱歉我的英文害您一個頭兩個大，還有實驗室裡的每位學長與學弟們，有大家的努力讓實驗室越來越茁壯！感謝產碩專班的好同學坤彥、雯謹、俊樺、家茵，因您們的存在，產碩專班的感情一天比一天更好，感謝產碩專班的助理雅珽，有您的提醒讓我們這群時常忘東忘西的傻學生都能夠順利畢業，感謝未來公司裡的Tom、Eason、東哥、David、Danny、Rola，在暑假實習的日子裡，承蒙您們的照顧，在此也要特別感謝Eason對本篇論文的支援，更得要感謝曾建超老師、胡誌麟老師、林順傑老師，在口試過程中給予我建議與指導，讓我的論文內容更臻完美，最後最後，我必須要感謝默默在我背後支持我的家人，有您們的支持讓我無後顧之憂，您們是最棒的家人，還有已經陪伴我六年多的女友思佩，雖然你修改我論文時很愛嫌東嫌西，但我知道這就是我們的生活樂趣，要感謝的人眞得很多，或許有些人我已忘卻了名字，但是我依舊會記得您的臉龐，眞的！有您們眞好，謝謝。

目　　　　錄

表 目 錄

圖　目　錄

Chapter 1

Introduction

A recommender system is like a specialized and specific information search engine, which tries to show the items that users might be interested in with users' and items' profiles. Generally speaking, most recommender systems are based on one of the following methods: content-based filtering, collaborative filtering, and hybrid method that combines the above two techniques. Content-based filtering recommender systems [1] make suggestions by selecting the features of the product information about consumers' preferences. Such systems always need users to check a variety of options about the products. Collaborative filtering recommender systems [2] make suggestions to customers by analyzing the previous users' collective experience about the products [3]. Such systems believe that the users who accept the products before, will also accept the similar products in the future. The hybrid recommender systems [4] combine the characteristics of the above two methods. Such systems use the past users' experience at the features of the products as data. After calculating and integrating the data, they make suggestions to the customers.

In this paper, we attempt to build a communication-style book recommender system "Bookle". It works just like you ask to staff what you want in the bookstore. Users could use Bookle to obtain the suggested books by describing the book content, instead of searching titles or abstract keywords. Our system is an informed recommender, which collected a lot of customer book reviews as data source. In addition, Our system is like a book content-orientation search engine, making suggestions from the book content that users describe. Users could use Bookle service for book recommendation as using Google Search for web page recommendation.

The main method of Bookle is similar to Collaborative Filtering. This approach requires a large number of reader ratings. Collaborative filtering systems use a collection of historical ratings data of n users on m products as input, collected by asking users to rate products [5]. Collecting such rating data need the users to spend time responding, and the values of ratings might not exactly provide reliable estimations of user preferences. Hence, the features of books are much difficult to define. An alternated approach is needed for Bookle to determine the quality of books. Hence, Bookle adds some concepts from Content-based filtering [6], obtaining the content of books from reader reviews. To a certain degree, the reviews could represent the summary of the book by readers. By the reviews, Bookle could speculate about the content of books and make the suggestions based on the content information.

In our recommender system "Bookle", we wish that the factors influence to the result of recommendations do not only from the content of reviews, but also the factor of reviewers' level of expertise in books, or the degree of reviewers preference for books. With this information, the system can measure the qualities of reviews. In the past, the common methods to measure article quality is to rate articles by users. For example, users can score the articles through the radio button or the star-rating section in a web page. However, this scoring model is easily manipulated by man-made. The inadequate number of rating will make the score too high or too low. And there is a another unfair case about articles rating. A commenter publishes his non-mainstream views seriously, but his articles can not obtain good evaluation. Because his opinions are different from the others. In fact, his articles still have the reference value. In conclusion, measuring the article quality by users is not a appropriate method.

After some discussion, we decide to use reading behaviors to determine the review quality. The trend of mobile reading is rising. The system can collect the information of reading behaviors from handhold devices. The qualities of different reviews are different. Someone wrote reviews for fun, but someone wrote reviews seriously. There is an approach to determine the qualities of product reviews by consumers level of expertise in the product [7]. This approach needs to record the profiles of customers for ranking their level of experience. Here we do not want to record any users' personal profile in this system, i.e., users do not need to create an account for sharing their reviews in Bookle. Everyone is anonymous in Bookle’s eye, and Bookle rates their review qualities by analyzing the reading behaviors.

Bookle collects the data source from users, and the part of the data including book information, reading behaviors, and personal review. After the server-side data integration, the results of “Text-mining process“ will be associated with the keyword list of books, and no other insignificant book information to be generated. After that the “Book-ranking process” sorts the related books for the keywords. Finally, users could use the service from the “Recommender process”. Because the recommender is based on filtering data from reviews, instead of the keyword search from book name or abstract (e.g., Google book search), Bookle could recommend the books which are much close to the users desired.

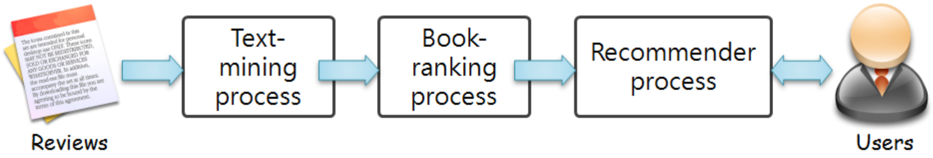


Figure 1.1: The process structure of Bookle recommender system.

In the following chapters, we will describe some related work and useful techniques about our research in chapter 2. Then we explain our idea and development scenarios in chapter 3. In chapter 4, we introduce the system architecture and explain the all components in the system workflow step by step. The implementation results and experiment are shown in chapter 5. Finally, we make a conclusion and mention some future works for our research in chapter 6.

Chapter 2

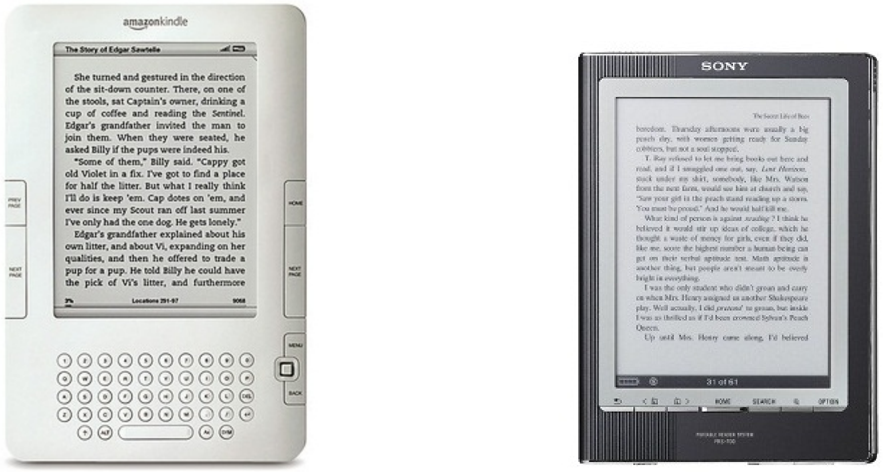
Related Work

It is difficult to define whether a book is worth to be recommended or not. Readers often do not know whether the book is recommended to their friends or not until they finish this book. Books do not like the normal products that have standard specifications (e.g., Consumer Electronics Products), they do not have the specific features to determine their quality before reading them. The common book recommender systems make suggestions by purchase experience. When people want to buy a book at a shopping website, they are interested in a section, "Customers Who Bought This Item Also Bought". But we do not know if those recommended books fit your need. The quality of the book content is hard to determine, and it is subjective by the different users. Users often discuss the quality of a book by sharing their reviews. Many readers also mention the contents of books when they share their feelings in their reviews. For the book content-orientation recommendation, we think that the book reviews are the useful information to predict the content of books.

When we think how to find a data source of the book reviews that is valid, another problem occurred in our mind: "Are the qualities of each review all equal?" [8] Some people wrote reviews about their feelings concisely, by using only two or three sentences. On the contrary, some people wrote their reviews detail, including the contents of books, their feelings, and so on. In intuition, the latter review has better quality. But, the review with more sentences possess the better quality? Some malice users would post some insignificant information to website message boards. Such information is most advertisement, and was repeated in multiple sites. It is difficult to determine the spam through the algorithm method. Most internet companies use artificial methods to do inspections for the spam. Hence, determine the review quality from the length of review content is not a good idea.

An informed recommender system based on consumer product reviews [7]. This informed recommender uses prioritized consumer product reviews to make recommendations. By using text-mining techniques, it maps each piece of each review comment automatically into an ontology. The ontology consists with two parts. One part is "Product quality" that represents the opinion of the provider's valuation of the product features. The other part is "Opinion Quality" that includes several variables to measure the opinion provider's expertise with the product. Hence, this approach also considers the quality of reviews by "Opinion Quality". However, the ontology of this approach is used for the products that have specific features, which is not suitable for books. To determine the opinion quality, the ontology needs additional space to record the user's profile and experience. In our work, Bookle do not store any user's personal profile or skill. Because Bookle want to determine the review quality by user reading behaviors, and who be the writer that is not necessary.

Some research investigate the relationship between reading behaviors and users' interests. A research considers this topic by recording user's behaviors including underline, highlight, circle, annotation and bookmark to capture user's interest precisely in E-learning system [9]. Another research considers this topic by two processes. First step is to measure that whether a article is interesting by users reading time period. And the second is to discover the match pattern from the articles in which a user is interested. Finally, this system can judge what articles would be interesting to the user [10]. A research attempts to find the interesting books for children from web logs information which including reading time period, and reading progress [11]. Hence, Bookle attempts to use reading behaviors to determine review quality. The loving degree for the books of readers would present in their reading behaviors. For example, the less time period the readers spent for reading a book means the higher degree they like the book. However, how could we obtain the reading behaviors from readers? Fortunately, mobile reading with handhold devices is a new trend. Users are unnecessary to bring many books everywhere for reading. They just need to download the electronic files of books to their handhold device.



(a) Amazon Kindle source: Amazon.com

(b) Sony PRS reader source: Sony.com

Figure 2.1: E-book readers

In many kinds of handhold devices, “E-book” device is the newer product and the most suitable device for reading. The characteristic of such device is the screen without backlight, just like watching the paper. Depending on this device, an E-book may be readable in low light environment. Many newer E-book devices have the ability to display motion, enlarge or change fonts, use Text-to-speech software to read the text aloud, search for key terms, find definitions, or allow highlighting bookmarking and annotation. The devices that utilize E-Ink can imitate the look and ease of readability of a printed work while consuming very little power, allowing continuous reading for weeks at a time. E-book is not similar to the LCD panel devices (e.g., Netbook or iPad) that is used for multimedia, but E-book is more suitable for reading text content. The well-known E-book devices on the market are Amazon Kindle, iRex iLiad, and Sony PRS series and so on.

With E-book devices, we attempt to make an universal device side library for recording reading behaviors and implement this library in it at first. In addition, we provide the function that can write the book reviews in E-book devices. When users finish reading books, they can write the book review immediately and the reviews would be one part of our data source. We could obtain the valid review data and reading behaviors together by using this approach [12]. Finally, we could determine the review quality by the information about users reading behaviors collected from E-book.

2.1 Automatic Keyword Extraction

Automatic keyword extraction is a technique to identify meaningful and representative fragments, or words. Keyword is the minimal unit to express the topic of many documents. Most automation applications use keyword extraction technique in unstructured documents, such as data mining, automatic answer, automatic filtering and so on. [13] In other words, keyword extraction is the basis and core technique for all documentation automations. Keyword extraction has been investigated in many different academic fields. One kind of keyword extraction was be studies by the viewpoint of linguistics. The research of linguistics from the perspective of linguistic analysis of the subject system and method of extraction, use of lexical knowledge, syntactic knowledge, semantic knowledge, and chapter subject knowledge extraction of different levels. Such research processes called "Information Extraction" and "Natural Language Processing". As Figure 2.2 shown, the data of readers' reviews is converted into useful information (keywords) after "Keyword Extraction Process".

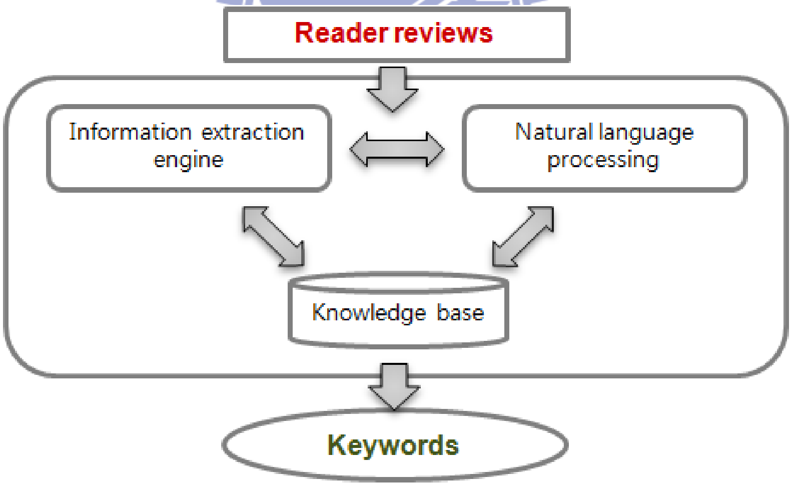


Figure 2.2: The keyword extraction process

2.1.1 Keyword Extraction Tool

In our research, we also use keyword extraction as preliminary work for the data of reviews. We attempt many different kinds of keyword extraction tools. We find the most appropriate one and use it in our research.

- Keyword Analysis Tool[1]: This tool is a web-base service. Users use it by assigning a web page URL, necessary keywords (optional), and stop keywords (optional) in its website. The results contain the keywords with only one word, the terms composed with multi-words named "keyphrases", and their frequency of "keywords" and "keyphrases". Keyword Analysis Tool is really powerful in keyword extraction. However, this tool is unsuitable for our work. Because it can only input data by assigning a URL, and without any open API for programmers. But our data of reviews was stored in database, and we do not intend to make web page for putting them temporarily. Hence, Keyword Analysis Tool is not an appropriate tool for us.

- Yahoo! Term Extraction[2]: The Term Extraction Web Service provides a list of significant words or phrases extracted from a larger content. It provides a open API for programmers. By this open API, we can post the content of reviews to the service, and obtain the results in XML or JSON format. The only weakness of this service is the results containing only keywords list without any other information about keywords' frequency or score. But it is still an useful tool for us to do keyword extraction, so we use this service in our work.

- Yahoo! CAS Web Service[3]: The CAS Web Service just like the Term Extraction Service, and it is more versatile. This service is developed by Taiwan Yahoo!. In addition to keyword extraction service, it also provide "Word Segmentation" service. Specifically, the results of keyword extraction have the score for each keyword. The score can make us to realize the importance of the keyword in the review. Here we do not use this service in our research at present stage, because this service was only used for Traditional-Chinese, and our research focuses on English review at beginning.

[1] Keyword Analysis Tool: http://seokeywordanalysis.com/

[2] Yahoo! Term Extraction Page: http://developer.yahoo.com/search/content/V1/termExtraction.html

[3] Yahoo! CAS Web Service: http://tw.developer.yahoo.com/cas/

Chapter 3

Research Topic and Methodology

Our recommender system is a communication-style service. Users request the recommendation by describing the book content that they want, and our system returns a suggestion of books to users. Similar to the collaborative filtering system, we need to collect sufficient reviews as source data. By those data source, we could build a book recommender database though a series of calculation process. Finally, the source data becomes the useful information for users.

3.1 System Conception

Bookle recommender system is a client-server base service. With the client-side library, any handhold devices could be the clients. The main job of clients is that collected the readers reading behaviors, saved the book reviews, and published them to Bookle server. Like a lot of on-line services, Bookle would be implemented as web-base service. The server side could be cut into two parts “front-end” and “back-end”. The back-end of Bookle server is responsible for accepting the information from clients and integrating them by a series of calculations. The front-end of Bookle server is a website that provides service interface for users. The users could describe the book content that they want and obtain the book recommendation list at this website. As Figure 3.1 shown, it expresses the system conception of Bookle service.

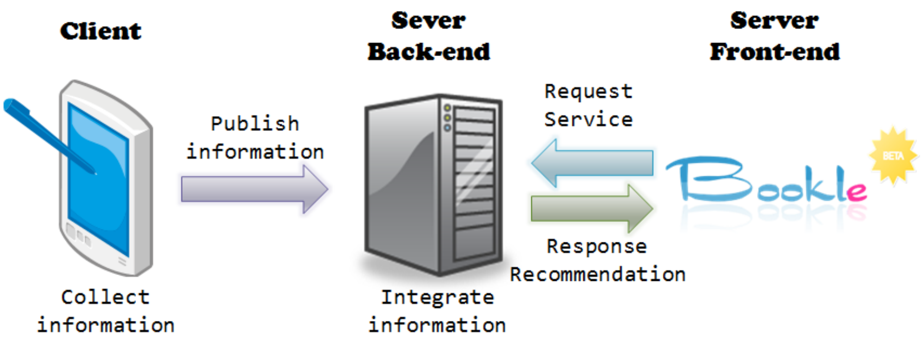


Figure 3.1: Bookle system overview

3.2 Client Information Collection

Bookle attempts to measure the quality of the reviews by reading behaviors. To protect readers' privacy, Bookle do not want to collect the information of who issued the reviews. In fact, Bookle only needs the data about how they read the book. We focus on the behaviors of reading, rather than the individual. For the quality of each review, we respect each reviewer's comment on the content. Readers may choose "bad" as a comment for the review, but the review is still valuable for our research.

To obtain the information of reading behaviors and reviews, Bookle needs to store them in the devices. We use the existing E-book functions to obtain useful data, such as bookmarking, adding to favorite, noting focus, etc. For more convenient maintenance, Bookle use a light database system to store the information in devices. When users read books, the users' reading behaviors would be saved in this storage unconsciously. The information that Bookle chooses to save in devices is described in the following chapter.

1. Book Name
2. ISBN (International Standard Book Number): an unique numeric commercial book identifier usually based upon the 10-digit or 14-digit numbers.
3. Date of Create at: The date when the book download to E-book device.
4. Date of Finish at: The date when the book read process arrives to 100%.
5. Bookmark: The bookmark numbers that reader created for the book.

6. Note: The note (paintings focus with circle or line) numbers that reader created for the book.

7. isFavorite: Is the book added into “my favorite”?

8. Review: The review that reader wrote for the book.

The above data is stored in the client-side device. When users finish writing reviews and publish them to the server-side machine, the above data would be transfered to the server together. At server side, we use these data of reading behaviors to determine the review quality. We will explain why Bookle choose to store above information in chapter 4 detail, and illustrate how could we determine the review quality with these data in section 4.1.1.

3.3 Server Operation Scenario

The main tasks for Bookle server-side are information integration and service presentation. For understanding our idea, we use the data of the book “The Lost Symbol” as an example to explain server operation. After collecting the book reviews and the reading behaviors pair by pair from E-book, Bookle would integrate the data and present service as the following scenario:

3.3.1 Key Extraction from the Review

First, Bookle obtains the key information from reviews by keyword extraction tool. With those keywords, Bookle could infer the book content and the feelings which readers read through the book. In Figure 3.2, there is an example that Bookle obtain the keywords from a review about book “The Lost Symbol”.

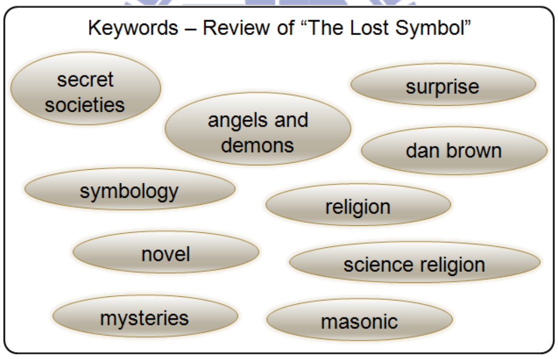


Figure 3.2: The keywords of a review of “The Lost Symbol”.

3.3.2 Keywords’ Weight Calculation

In next, Bookle calculates the weight of each keyword extracted from the reviews in one book according to their occurrences and the review quality. The weight means the keyword importance to the book. When a keyword often appear in different reviews, it will have a higher weight. In other words, if a keyword with high weight that is expressed people often referred to it in their reviews. It has closer relationship for this keyword to the book. The different review qualities help us to set unequal influence to the same keywords from different reviews. As Figure 3.3 shown, there are the weights of the keywords from all reviews about book “The Lost Symbol”.

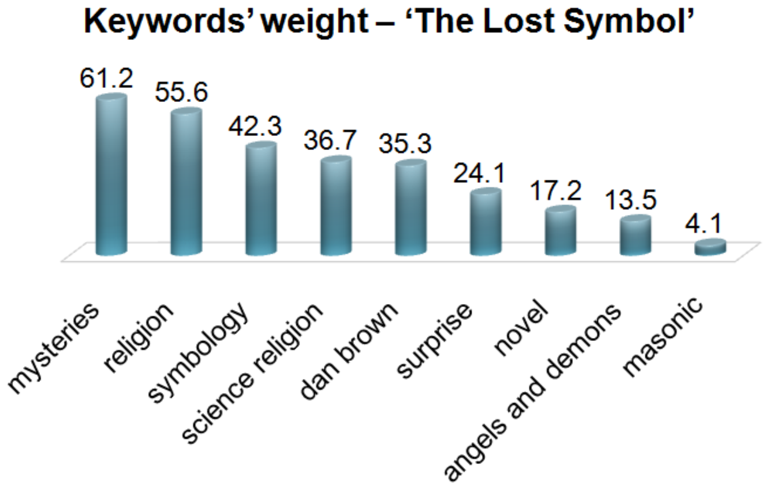


Figure 3.3: The weight of the keywords in the book “The Lost Symbol”.

3.3.3 Discovering the related Books according to the Keywords

This step is in a reverse thinking. After Bookle obtain the weight of every keyword in the books, Bookle start to obtain the related books for each keyword from the weight. For each keyword, the top-K values of keyword weight are discovered from all books, and stored to the database. Although the keywords are the same, but they are related to K different books. Hence, Bookle can obtain the most relevant K books for every keyword. In Figure 3.4, it shows the related books about keyword “mystery”.

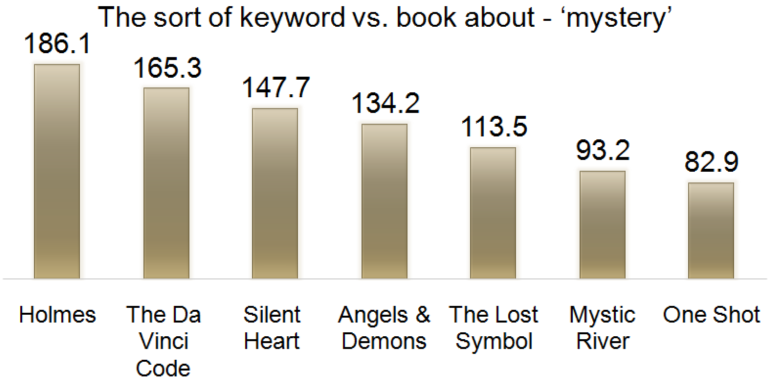


Figure 3.4: The books related to the keyword “mystery”.

3.3.4 Presentation of Bookle Service

When users give a query to the service by describing the book content. Bookle parse the description and obtain the keywords from the query. With these keywords, Bookle find the books from the relevant books of keywords created at previous step. As Figure 3.5 shown, the books is related to description, “The mystery suspense thriller”.

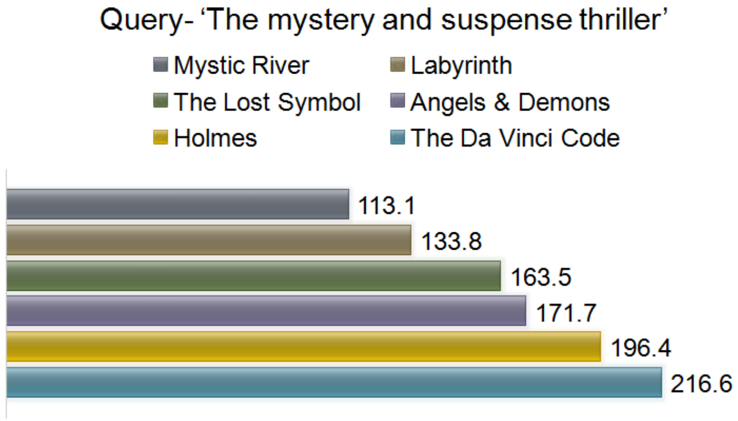


Figure 3.5: The results of the description “The mystery and suspense thriller”.

Chapter 4

Server Framework and Algorithm

Most of the recommender systems would be conducted modularly for more convenient maintenance and management, and Bookle is no exception. Bookle server could be generally divided into two parts. One part is the front-end for user service and communication with client, and the other part is the back-end for a series of calculation process. The following of this chapter will explain the Bookle server architecture and workflow in detail.

4.1 The Back-end of Server

The main task of the back-end is to convert the review data into useful information. The data which Bookle server collects from clients (E-book Users) needs to go through four main processing modules as Figure 4.1 shown.

- Text-mining: To obtain keywords from reviews by keyword extraction service. (Yahoo! Term Extraction Service)
- Review-rating: To integrate the reading behaviors and to rate reviews quality.
- Weight-calculating: To calculate the weight of each keyword related to books.
- Book-ranking: To rank the related books for each keyword by keyword weight.

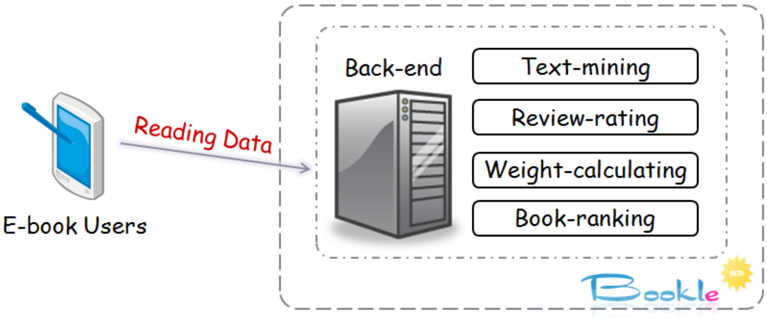


Figure 4.1: The Architecture of Server Back-end

4.1.1 The Workflow of Back-end

For converting the review data into useful information, the back-end of Bookle server runs the calculus processing as following workflow:

Step 1 : Extract keywords from each review. Bookle utilizes "Yahoo! Term Extraction tool" (mentioned in 2.1.1) to extract the representative keywords from reviews and store them into server database. If keywords is more than 20, Bookle records the first 20 keywords as Figure 4.2 shown.

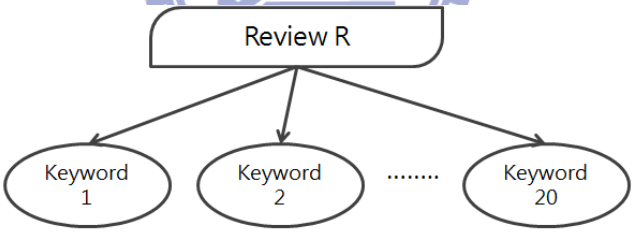


Figure 4.2: Keywords Extraction

Step 2-A : Calculate the average value of reading behaviors for each book. To rate review quality, Bookle acquires three different average values from reading behaviors. The first value is "Average Experience Period" that is a time period in days as the unit. The "Experience Period" is the difference of $Date\_of\_Finish\_at$ and $Date\_of\_Create\_at$ (mentioned in 3.2). The value of "Experience Period" does not mean the actual reading time that user spent for a book. The value could be the best indicator for measuring how users love the book. When readers are very interested in a book, they would desire to go through whole book as soon as possible. The second value is the amount of "Average Bookmarks" of the book. Users could set a bookmark at specific page in E-book for recording reading progress or just making a mark. The amount of bookmarks is also one of criteria for rating review quality. The last one is the amount of "Average Notes" in a book. Users could make the section that they feel interested in with a circle or an underline in pages. The amount of notes represents how seriously users read the book, and it is also a good reference for rating review quality.

$$ExperiencePeriod(EP) = Date\_of\_Finsih\_at - Date\_of\_Create\_at \tag{4.1}$$

If there are n reviews for a book, let R_i be the review i. Let EP_i, BM_i, and $Note_i$ represent the Experience Period, Bookmark number, and Note number respectively. The formula for calculating the average value of reading behaviors can be expressed as following:

- Average Experience Period

$$avg.EP = \frac{\sum_{i=1}^{n} EP_i}{n} \tag{4.2}$$

- Average Bookmark Number

$$avg.BM = \frac{\sum_{i=1}^{n} BM_i}{n} \tag{4.3}$$

- Average Note Number

$$avg.Note = \frac{\sum_{i=1}^{n} Note_i}{n} \tag{4.4}$$

Step 2-B : Measure the quality for each review. The main assessment method bases on the data about reading behaviors. There are four criteria for Bookle to measure the review quality.

1. Base Quality(BQ): The "Base Quality" is measured by the value about "Experience Period". This is the most obvious indicator to determine the degree how readers love one book, and this value is regarded as the most influential variables of review quality. When a reader likes a book very much, the value of Experience Period will be very small, even though he or she is in a busy life. If the Experience Period that someone spent on a book is ten times faster than the average value, the value is scored full marks from "Base Quality". Bookle

calculates the Base Quality by the ratio of average Experience Period and personal Experience Period. The Base Quality value of a review is calculated as Formula 4.5 shown.

$$BaseQuality(BQ) = \frac{avg.EP}{PersonalEP} \tag{4.5}$$

2. Bookmark Quality(BmQ): The value of “Bookmark Quality” is derived from the amount of bookmarks. In the process of how to decide the value of Bookmark Quality, Bookle consider the relationship about the Experience Period and bookmark numbers. For example, Kevin has a lot of leisure time to do things in which he interested. Hence, he could read his favorite books in a short period of time. From his reading behaviors, the “Experience Period” is seven times faster than the average and he made few bookmarks. For another example, Alice is a busy office worker, and she must make good use of spare time on her interests. Hence, it took her longer time period than Kevin to read through the same book. From her reading behaviors, the “Experience Period” is three times faster than the average and she made many bookmarks. If Bookle only consider the value of “Experience Period”, the fronter has better review quality. However, Alice made more efforts on reading the book than Kevin, so the later should have better review quality. It should be realized that Bookle need to provide the additional quality score from the data of bookmarks for the later. Generally speaking, a review with more bookmarks is valued with higher Bookmark Quality in the case of the same Experience Period. To obtain the Bookmark Quality, Bookle calculates the quotient of personal bookmark numbers and average bookmark numbers for relative comparison. The complete value of “Bookmark Quality” is the calculation about “Experience Period” and “Bookmark Number” as Formula 4.6 shown.

$$BookmarkQuality(BmQ) = \frac{avg.EP}{PersonalEP} * \frac{PersonalBM}{avg.BM} \tag{4.6}$$

3. Note Quality(NQ): The expression of Note is a circle or an underline on pages for making a key block. The readers who make notes while reading are usually more diligent, but personal reading habits and types of books are also factors which should be taken into consideration. It generates more positive effects in review quality if more notes are drawn in a book. As Formula 4.7 shown,

Bookle measures Note Quality by comparing personal note numbers with the average note numbers of a book. If personal note numbers are ten times more than the average value, it would be scored full marks in this criteria.

$$NoteQuality(NQ) = \frac{PersonalNote}{avg.Note} \tag{4.7}$$

4. Favorite Quality(FQ): If readers put a book into virtual bookcase "My Favorite", the review quality would be rewarded extra bonus.

In conclusion, the total value of review quality is the sum of the above four criteria as Formula 4.8 shown. We can adjust the constant terms of the four criteria dynamically by situation. The complete measurement of review quality is expressed as Figure 4.3 shown.

$$ReviewQuality = \alpha BQ + \beta BmQ + \gamma NQ + \delta FQ \tag{4.8}$$

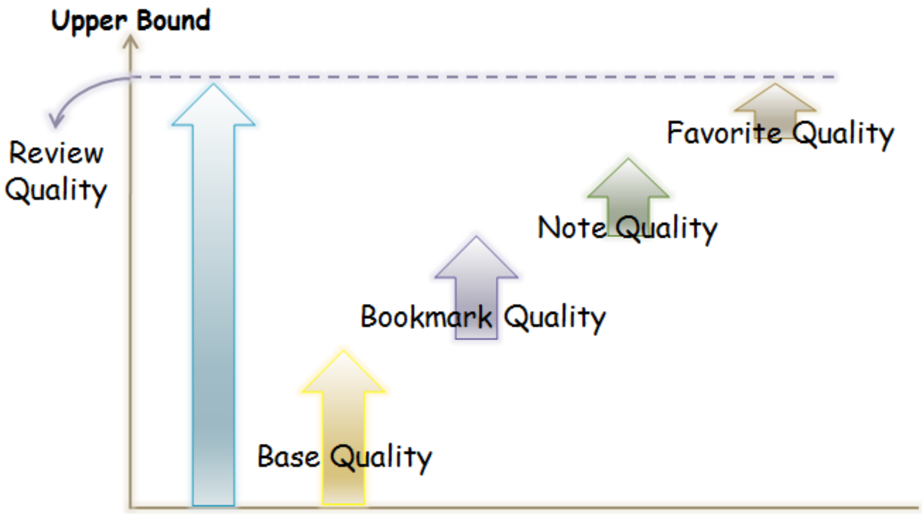


Figure 4.3: Review Quality Measurement

Step 3 : Calculate keyword weight for each keyword in each book. Keywords are extracted from reviews of books at Step 1, and Bookle would assign a weight for every keyword as Figure 4.4 shown. Bookle calculates the weight of each keyword by occurrences and quality of reviews which the keywords occurred. Same keywords keywords might reoccur in different reviews of one book, and the quality values of different reviews combine into the weight of the keyword. In generally, the more the keyword occurs, the better weight this keyword is scored. Bookle obtains the weight of a keyword by the sum of the review quality value from which the keyword is extracted. For keyword T of the book B, if keyword T reoccurs in n reviews R_n and the review quality of R_n is Q_n, the weight of keyword T for this book B is the sum of Q_n as Formula 4.9.

$$Weight(T) = \sum_{i=1}^{n} Q_n \tag{4.9}$$

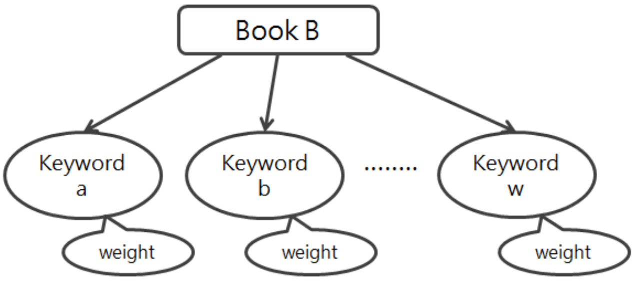


Figure 4.4: The Weight for Keywords

Step 4 : For each keyword, rank the books that related to the keyword. At step 3, Bookle obtains information about keyword weights of all books. At this step, Bookle utilizes the information to rank books for keywords. The same keywords might have relation with different books, because readers have the same experience between the books. The same keywords in different books would be weighted in different value. Bookle ranks books for each keyword by sorting the top-K weights. After this step, Bookle obtains information about the top-K books for each keyword as Figure 4.5 shown.

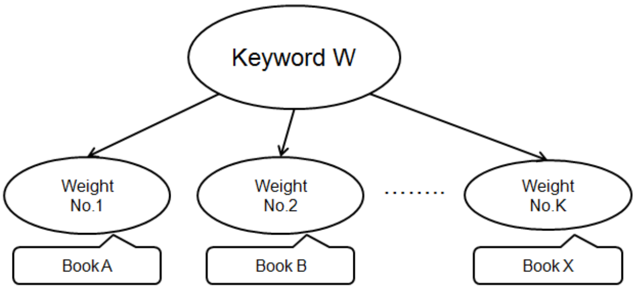


Figure 4.5: The Books Related to Keywords

Finally, Bookle can make the suggestion by the information from Step 4 when users request for recommender service. The information about books related to keywords could help Bookle to speculate what books users want. This step is also the final step of back-end in Bookle server.

4.2 The Front-end of Server

The main function of the front-end is regarded as a user interface or programmer interface. There are two main tasks in front-end, one is the website service [14], and the other is the restful API for the unified usage.

- Website Service: Bookle designs a website for recommender service. Users could describe book contents in which they are interested. And then, Bookle would show a recommendation list of related book sorted by "Bookle Score".
- Restful API: Bookle provides two restful API for programmer usage. The programmer could develop all kinds of service through the two API. Bookle restful API requests should be submitted using an HTTP POST request rather than GET.
  1. Review Acceptor: Accept information about user reading behaviors and reviews from handhold device (E-book) and store them in server database. The information must include book ISBN, Date_of_Create_at, Date_of_Finish_at, bookmark numbers, note numbers, isFavorite message, and review (mentioned at chapter 3.2).
  2. Recommender: Response the book recommendation query. The POST message must include book description. The result format of Recommender could be XML or JSON by the parameter "type" in POST.

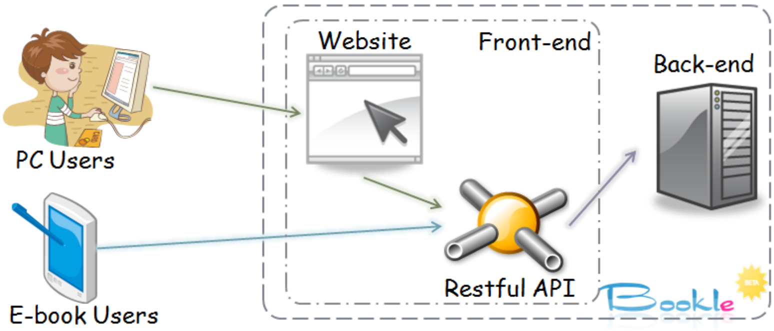


Figure 4.6: The Architecture of Server Front-end

4.2.1 The Workflow of Recommender

When users submit the book description to request Bookle recommender service. Bookle would deal with the description as following steps:

Step 1 : Users request Bookle recommender service by describing the book content in which they are interested.

Step 2 : Extract keywords from the user description by term extraction tool (mentioned at chapter 2.1.1).

Step 3 : Select the related books from the "Book Ranking for Keywords" table by the extracted keywords.

Step 4 : Cross compare to find the same book and sum up the weight of keywords as Bookle Score.

Step 5 : Reply a recommendation list to the users with top-10 books in Bookle Score.

For more convenience, the results of recommendation could be packed in the XML or JSON format. Programmers can show the result of recommendation in many different ways easily. Programmers could develop their services at handhold devices or any application environments. At present, Bookle provides the base service for users by the website.

Chapter 5

Implementation and Experiment

5.1 Client Storage

Because the power of handhold devices is limited, Bookle use a light and convenience database system named “SQLite” [1]. Dislike general database systems, SQLite is not a client-server based database. Whole database (including definition, tables, index, and data) were existed in single file. Because handhold devices only have limited computing ability and battery capacity, “SQLite” is the best choice of database system for Bookle client-side.

5.2 Communication

When readers read through a book and write reviews, they can publish the review to the Bookle server. Meanwhile, the information of reading behaviors would also be transfered to Bookle server. For transferring those information to server, Bookle make the communication API base on “libcURL” [2] library. “libcURL” is a free and easy-to-use client-side URL transfer library, supporting FTP, HTTP, HTTPS, SMTP, POP3, etc. The main task for this communication API is responsible for submitting review information from client to server and dealing with the responses. Relatively, there are some restful APIs in Bookle server-side for accepting requests from client. As the Figure 5.1 shown,

[1] SQLite reference page: http://www.sqlite.org/index.html

[2] LibCURL reference page: http://curl.haxx.se/

this is the concept diagram about client-server communication.

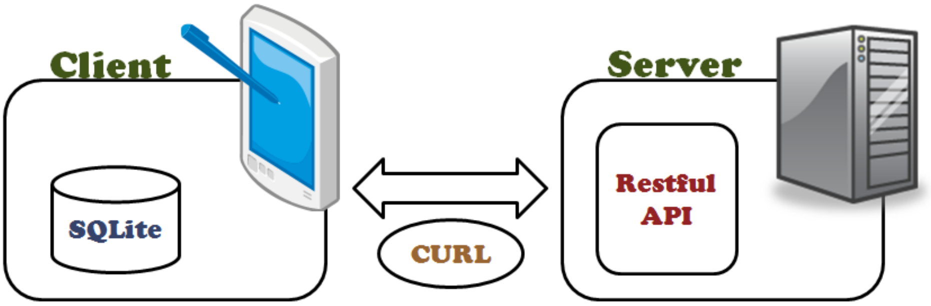
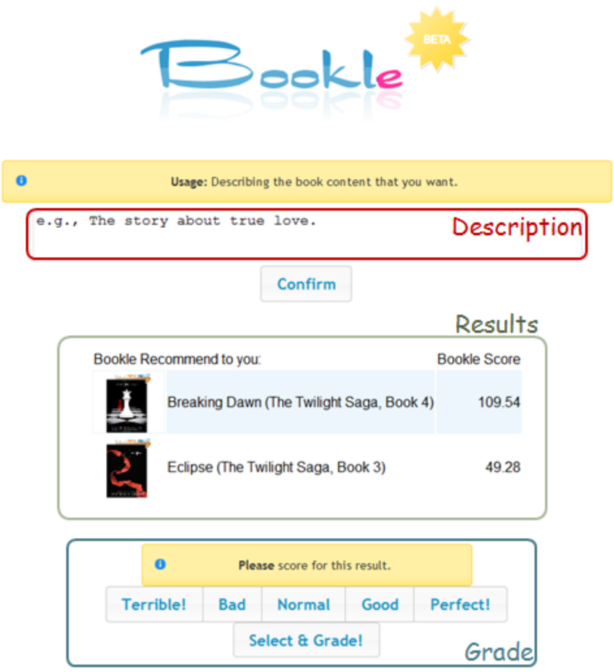


Figure 5.1: The communication of client-server

5.3 Results Presentation

As Figure 5.2 shown, it is a snapshot of Bookle website that provides recommender service. Users can describe book contents they want in the text field, and click the button "Confirm" to request recommender service. The results of recommendation would be showed at the bottom of this page. For example, someone writes the description "The story about true love." and request to the service. The Bookle service responses a recommendation list with two books, one is "Breaking Dawn", and the other is "Eclipse". In this recommendation list, Bookle would provide an indication to express the degree of consistency for the description by "Bookle Score". This score is calculated by the weight of keywords extracted from description. The more reviews are related to one book, the better score the book gets. In other words, this score could be regarded as the popularity of books. Books with higher score indicating more relevant to the user needs. Users can reference this information to discover the books in which they are interested.

Most recommender systems use the users transaction records as data source. The most common type of this recommender service is "Customers Who Bought This Item Also Bought". When you go shopping at online store, you can often see a block about this service at the website. However, it usually can not provide what you really want. Actually, it provides information about popular products. When a product sells better, it is more possible to appear in this block. For example, Andy wants to buy some love stories at online store (e.g., Amazon.com). He decides to buy the love story "Breaking Dawn",



Usage: Describing the book content that you want.

e.g., The story about true love.

Bookle Recommend to you:	Bookle Score
Breaking Dawn (The Twilight Saga, Book 4)	109.54
Eclipse (The Twilight Saga, Book 3)	49.28

Please score for this result.

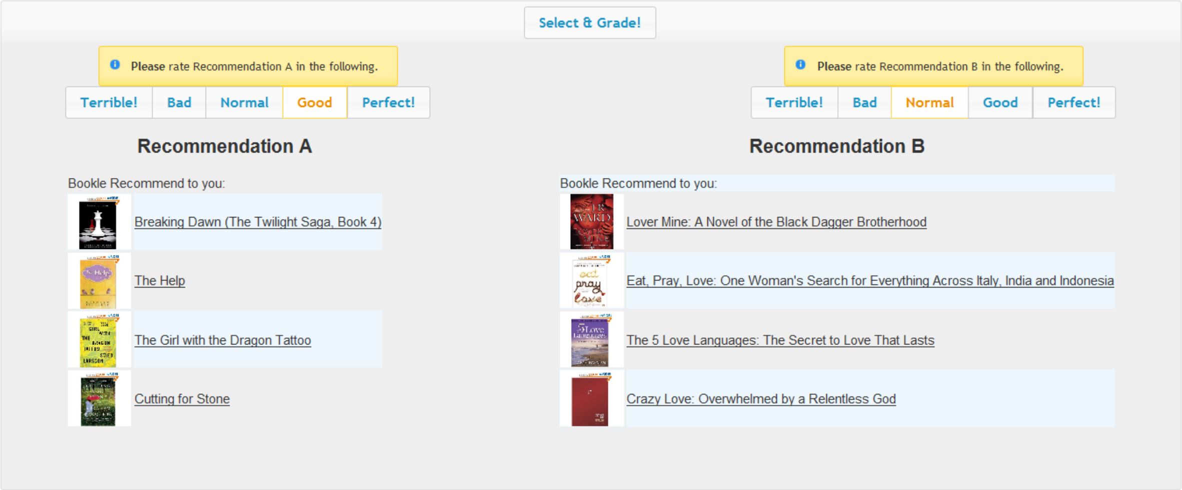
Figure 5.2: Bookle Website Demo

and references to the block named “Customers Who Bought This Item Also Bought” for more books at the same web page. He sees the book “Harry Potter and Deathly Hallows” in this block, but the book is a fantasy and mystery fiction.

Different from traditional recommender services, Bookle utilizes reader reviews as data source. The results of recommendation are the most relevant books about user’s description, but it does not mean that “Customers Who Bought This Item Also Bought” service should be discarded. This service offers the related products and popular products information, and Bookle service like a book content search engine which lists books that you want to read. The two systems are designed on different data source and utilizes different methods, but work for similiar purpose. However, the two systems have their own characteristics and advantages.

5.4 Experiment

In experiment, we utilizes a common search method to be compared with Bookle. When users describe the book content in experiment web page, they will obtain two kinds of results. One result is recommended by Bookle service, and the other is recommended by keyword search from book title or abstract, which is called default method temporarily. If one party does not response any result, it will get 0 points. As Figure 5.3 shown, it is a snapshot of the experiment web page. Users can grade the results from the ratio buttons. According to the satisfaction of results, users can score for results from 1 to 5. The information recorded in the experiment including description strings, the scores for Bookle, and the scores for default method.



Please rate Recommendation A in the following.

Bookle Recommend to you:

- Breaking Dawn (The Twilight Saga, Book 4)
- The Help
- The Girl with the Dragon Tattoo
- Cutting for Stone

Please rate Recommendation B in the following.

Bookle Recommend to you:

- Lover Mine: A Novel of the Black Dagger Brotherhood
- Eat, Pray, Love: One Woman's Search for Everything Across Italy, India and Indonesia
- The 5 Love Languages: The Secret to Love That Lasts
- Crazy Love: Overwhelmed by a Relentless God

Figure 5.3: The Snapshot of Experiment Page

5.4.1 Data Source

Because we can not obtain large amount of data from E-book devices at this stage, we need to use the existing resources. The main data source can be divided into two parts. One part is the data of book reviews. For obtaining large amount of review data, we crawl the book reviews from Amazon.com as data source. And the other part is the data of reading behaviors. On existing resource, it is hard to acquire the data of reading behaviors. We need to make some data of reading behaviors appropriately. The artificial data of reading behaviors contained “Experience Period”, “Bookmark numbers”, “Note numbers” and “isFavorite”. We utilize certain specific rules to generate the data of reading behaviors.

- Book Reviews: The reviews of top-100 books from Amazon.com in April and May, and the amount of reviews of a book is more than 50. Currently there are 50 books and more than 18,000 reviews in the server database.

- Reading Behaviors

  1. Experience period($Date\_of\_Creat\_At$): To generate a period value P by Normal Distribution Random Number. $ExpectedValue = PageNumber/15$ and $StandardDeviation = 1$.
     $Date\_of\_Create\_At = Date\_of\_Finish\_At - P$ (Unix Timestamp format)
  2. Experience period($Date\_of\_Finish\_At$): It is the date that readers posted reviews at Amazon.com in UNIX Timestamp format.
  3. Bookmark Number: To generate a random integer by Normal Distribution Random Number. $ExpectedValue = PageNumber/15$ and $StandardDeviation = 1$.
  4. Note Number: To generate a random integer by Normal Distribution Random Number. $ExpectedValue = PageNumber/5$ and $StandardDeviation = 1$.
  5. isFavorite: Randomly to 0 or 1.

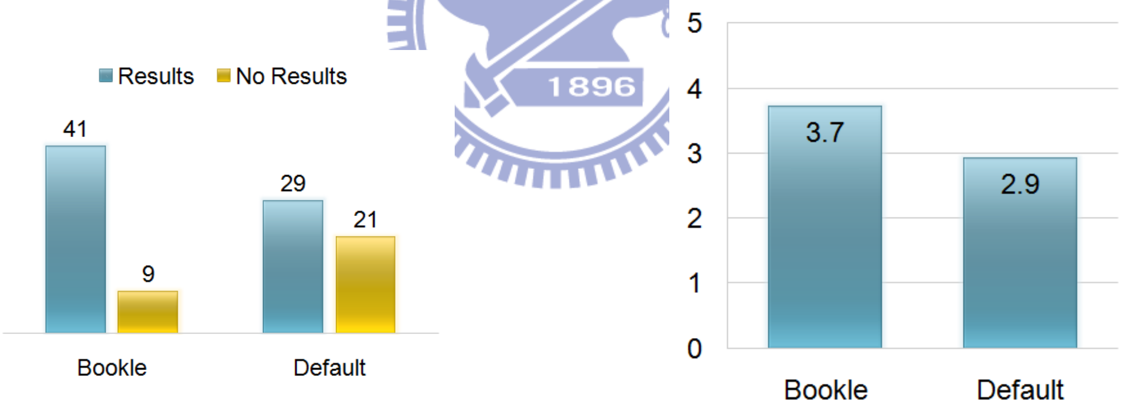
5.4.2 Experiment Results

In the experimental process, we invited the 6 volunteers to rate our service. They tested a total of 50 descriptions to our system, and we use the parameter settings for measuring review quality as Table 5.1 shown.

Table 5.1: The Criteria Ratio of Review Quality in Experiment

	Base Quality	Bookmark Quality	Note Quality	Favorite Quality
Percentage	55%	25%	15%	5%

As the Figure 5.4(a) shown, It is often occurred in default method that no results is responded because the keywords of the description can not be found from book title or abstract. Comparing to the situation of results, the average score of Bookle is sightly better than that of default method in Figure 5.4(b). Generally speaking, most of the volunteers think the results of Bookle are more adequate to the requirements than that of default method.



(a) Whether the Results for Description

(b) Average Scores of Results

Figure 5.4: Experiment Results

Chapter 6

Conclusion

Bookle is a recommender system for books. Books are special products, it is hard to rate the quality of books by specific features. Bookle attempts to make suggestion in another way that users obtain book recommendation by describing book content they wanted. Bookle utilizes the reviews as basic data source and extracts key elements from reviews by keyword extraction technique. Research about using reviews in recommender systems is still in its infancy. To the best of our research, this is the first attempt to build a recommender system for books based on reader reviews in free-form text. To be more credible, Bookle measures the review quality by reading behaviors. Although Bookle already takes many elements into consideration, there are still a lot could be improved. First, so far there are no incentives for users to write reviews for Bookle. We suggest network benefits for users willing to share their reviews, so the social network connection with Bookle should be built as soon as possible. Users would write comments because they want to share their opinions with friends. The second one is the importance of a keyword in one review. Bookle thinks that the keywords of a review have the same importance, but it should distinguish the importance of individual keyword in one review. Some keyword extraction services could provide the importance degree of keywords in results. The last but not the least, Bookle can consider more criteria of reading behaviors, for example, the max time period which users read the book and how many times users flip over pages in a continuous reading. Taking those criteria into consideration could enhance the reliability of reviews and avoid improper operation of users effectively. With the three improvements mentioned above, Bookle would gradually got enhanced in the future.

Bibliography

[1] G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.

[2] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” pp. 291–324, 2007.

[3] D. Billsus and M. You, “Learning collaborative information filters,” *Proc. 15th Int’l Conf. Machine Learning*, pp. 46–54, 1998.

[4] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modeling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.

[5] W. Yang, Z. Wang, and M. You, “An improved collaborative filtering method for recommendations’ generation,” in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 5, pp. 4135–4139, 2004.

[6] P. Melville, R. J. Mooney, and R. Nagarajan, “Content-boosted collaborative filtering for improved recommendations,” *Eighteenth National Conference on Artificial Intelligence*, pp. 187–192, 2002.

[7] S. Aciar, D. Zhang, S. Simoff, and J. Debenham, “Informed recommender: Basing recommendations on consumer product reviews,” *IEEE Intelligent Systems*, vol. 22, no. 3, pp. 39–47, 2007.

[8] D. Cosley, S. K. Lam, I. Albert, J. A. Konstan, and J. Riedl, “Is seeing believing?: how recommender system interfaces affect users’ opinions,” in *CHI ’03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 585–592, ACM, 2003.

[9] Y. Liang, Z. Zhao, and Q. Zeng, “Mining user’s interest from reading behavior in e-learning system,” in *Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on*, vol. 2, pp. 417–422, 2007.

[10] M. Morita and Y. Shinoda, "Information filtering based on user behavior analysis and best match text retrieval," in *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 272–281, Springer-Verlag New York, Inc., 1994.

[11] R. Chen, A. Rose, and B. B. Bederson, "How people read books online: mining and visualizing web logs for use information," in *ECDL'09: Proceedings of the 13th European conference on Research and advanced technology for digital libraries*, pp. 364–369, Springer-Verlag, 2009.

[12] W. Woerndl, C. Schueller, and R. Wojtech, "A hybrid recommender system for context-aware recommendations of mobile applications," in *ICDEW '07: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pp. 871–878, IEEE Computer Society, 2007.

[13] A. M. Ráez and R. Steinberger, "Why keywording matters," *High Energy Physics Libraries Webzine*, 2004.

[14] P. Kazienko, "Web-based recommender systems and user needs –the comprehensive view," in *Proceeding of the 2008 conference on New Trends in Multimedia and Network Information Systems*, pp. 243–258, IOS Press, 2008.

[15] J. A. Konstan, "Introduction to recommender systems," in *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1373–1374, ACM, 2008.

[16] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 167–174, ACM, 2005.

[17] M. Degemmis, P. Lops, and G. Semeraro, "A content-collaborative recommender that exploits wordnet-based user profiles for neighborhood formation," *User Modeling and User-Adapted Interaction*, vol. 17, no. 3, pp. 217–255, 2007.

[18] M. Elahi, "Context-aware intelligent recommender system," in *IUI '10: Proceeding of the 14th international conference on Intelligent user interfaces*, pp. 407–408, ACM, 2010.

[19] M. Balabanović and Y. Shoham, "Fab: content-based, collaborative recommendation," *Commun. ACM*, vol. 40, no. 3, pp. 66–72, 1997.

[20] P. Massa and P. Avesani, "Trust-aware recommender systems," in *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pp. 17–24, ACM, 2007.

[21] T. Tran, "Designing recommender systems for e-commerce: an integration approach," in *ICEC '06: Proceedings of the 8th international conference on Electronic commerce*, pp. 512–518, ACM, 2006.

[22] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *RecSys '08: Proceedings of the 2008 ACM conference on Recommender systems*, pp. 335–336, ACM, 2008.

[23] Liu and Ziming, "Reading behavior in the digital environment: Changes in reading behavior over the past ten years," *Journal of Documentation*, vol. 61, pp. 700–712, January 2005.

[24] K. O'Hara and A. Sellen, "A comparison of reading paper and on-line documents," in *CHI '97: Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 335–342, ACM, 1997.