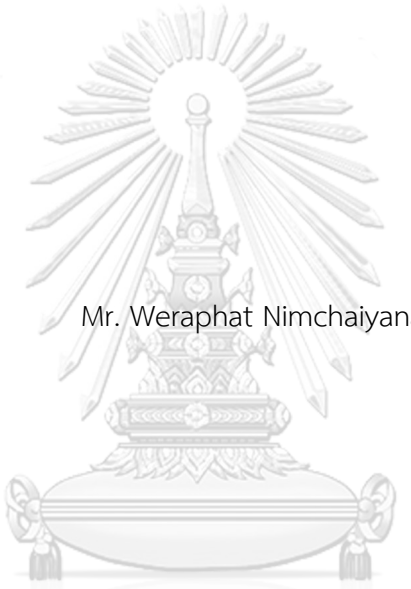
A HOTEL HYBRID RECOMMENDATION METHOD BASED ON CONTEXT-DRIVEN USING LATENT DIRICHLET ALLOCATION

Mr. Weraphat Nimchaiyanan



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science and Information Technology
Department of Mathematics and Computer Science
Faculty of Science
Chulalongkorn University
Academic Year 2018

วิธีการแนะนำโรงแรมแบบผสมโดยพิจารณาบริบทด้วยการจัดสรรแฝงของดีรีเคล

นายวีรภัทร นิ่มไชยนันท์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

Thesis Title	A HOTEL HYBRID RECOMMENDATION METHOD BASED ON CONTEXT-DRIVEN USING LATENT DIRICHLET ALLOCATION
By	Mr. Weraphat Nimchaiyanan
Field of Study	Computer Science and Information Technology
Thesis Advisor	Assistant Professor Dr. SARANYA MANEEROJ, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

..... Dean of the Faculty of Science

()

THESIS COMMITTEE

..... Chairman

(Associate Professor Dr. PERAPHON SOPHATSATHIT, Ph.D.)

..... Thesis Advisor

(Assistant Professor Dr. SARANYA MANEEROJ, Ph.D.)

..... External Examiner

(Assistant Professor Dr. Saichon Jaiyen, Ph.D.)

วีรภัทร นิ่มไชยนันท์ : วิธีการแนะนำโรงแรมแบบผสมโดยพิจารณาบริบทด้วยการจัดสรรแฝงของดีรีเคล. (A HOTEL HYBRID RECOMMENDATION METHOD BASED ON CONTEXT-DRIVEN USING LATENT DIRICHLET ALLOCATION) อ.ที่ปรึกษาหลัก : ผศ. ดร.ศรันญา มณีโรจน์

ระบบการแนะนำเข้ามามีบทบาทสำคัญในการช่วยผู้ใช้งานในการหาแนะนำข้อมูลหรือสินค้าที่ผู้ใช้งานต้องการ โดยทั่วไประบบการแนะนำจะมีการใช้คะแนนของผู้ใช้งานกับเทคนิคการกรองแบบอิงเนื้อหา(content-based filtering)หรือเทคนิคการกรองแบบอ้างอิงกิจกรรมความร่วมมือ(collaborative filtering) เพื่อแนะนำข้อมูลแก่ผู้ใช้งาน ในปัจจุบันการใช้แค่เพียงคะแนนของผู้ใช้งานอย่างเดียวนั้นไม่เพียงพอต่อการแนะนำข้อมูลต่าง ๆให้กับผู้ใช้งาน ดังนั้นข้อมูลตามบริบท(contextual information), บริบทที่ถูกขับเคลื่อน(context driven) และการจัดสรรแฝงของดีรีเคล(Latent Dirichlet Allocation) จึงถูกนำมาใช้เพื่อพัฒนาระบบการแนะนำ ในโครงงานมหาบัณฑิตนี้มีจุดประสงค์เพื่อนำเสนอวิธีการแนะนำโรงแรมแบบผสมโดยพิจารณาบริบทด้วยการจัดสรรแฝงของดีรีเคลซึ่งในงานวิจัยนี้มีขั้นตอนดังนี้เริ่มจากหาคะแนนของผู้ใช้งานที่ผู้ใช้งานยังไม่เคยใส่คะแนนมาก่อนจากตารางผู้ใช้งานกับโรงแรมด้วยการใช้การจัดสรรแฝงของดีรีเคล จากตารางที่ถูกเติมเต็มด้วยขั้นตอนแรกค่าที่ได้ถูกนำไปใช้เพื่อหาเพื่อนของผู้ใช้งานเป้าหมายโดยใช้การกรองแบบอ้างอิงกิจกรรมความร่วมมือ และสุดท้ายทำการเลือกโรงแรมที่จะแนะนำแก่โดยผู้ใช้งานเป้าหมายโดยคำนึงถึงความชอบในปัจจุบันของผู้ใช้งานเป้าหมายที่หาได้จากการใช้รีวิวของโรงแรมที่เพื่อนของผู้ใช้งานเป้าหมายแนะนำซึ่งในขั้นตอนนี้จะมีการนำบริบทที่ถูกขับเคลื่อนมาใช้ การประเมินประสิทธิภาพของงานวิจัยนี้ทำโดยการนำวิธีการของงานวิจัยนี้ไปเปรียบเทียบกับวิธีอ้างอิงที่มาจากวิธีที่ใช้เทคนิคการกรองแบบอิงเนื้อหากับการจัดสรรแฝงของดีรีเคลและวิธีที่ใช้ทคนิคการกรองแบบอ้างอิงกิจกรรมความร่วมมือกับการจัดสรรแฝงของดีรีเคลด้วยการใช้ตัวชี้วัดที่เรียกว่า Normalized Discounted Cumulative ซึ่งผลที่ได้พบว่าวิธีของงานวิจัยนี้สามารถแนะนำและจัดลำดับโรงแรมที่ตรงตามความชอบของผู้ใช้งานเป้าหมายมากที่สุดเมื่อเทียบกับวิธีอ้างอิงอื่น

สาขาวิชา	วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ	ลายมือชื่อนิสิต .....
ปีการศึกษา	2561	ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

5972616723 : MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
KEYWORD: content-based filtering, collaborative filtering, hotel recommendation, Latent Dirichlet Allocation, context driven

Weraphat Nimchaiyanan : A HOTEL HYBRID RECOMMENDATION METHOD BASED ON CONTEXT-DRIVEN USING LATENT DIRICHLET ALLOCATION. Advisor: Asst. Prof. Dr. SARANYA MANEEROJ, Ph.D.

Recommender systems play an important role in helping users find items that they want. Normally, ratings are used in content-based filtering (CBF) and collaborative filtering (CF) for recommendation. However, only ratings are not enough for recommendation. Thus, contextual information, context driven and Latent Dirichlet Allocation (LDA) are used to improve recommendation. Also, the context of individual user has changed in the timeline (context-driven). In this work, a hotel hybrid recommendation method (CF+CBF) based on context-driven using LDA is proposed. Firstly, we find missing user ratings of user-hotel rating matrix by applying LDA on user ratings in order to get predicted score of hotels for the target user. Secondly, we find a group of users similar to the target user (neighbors). Then, we apply context-driven to recommend hotels that meet current interest of the target user. To evaluate the proposed method, we compare our proposed methods either CBF or CF integrating with LDA by measuring nDCG. The result shows that our proposed method outperforms all comparable methods in result accuracy.

Field of Study:	Computer Science and Information Technology	Student's Signature .....
Academic Year:	2018	Advisor's Signature .....

ACKNOWLEDGEMENTS

This thesis could not accomplish without the help and support from the faculty and staffs during the implementation. I am grateful for the following assistance and would like to thank all of those who made it possible for me to complete this thesis.

Firstly, I am very much obliged and grateful to my research advisor, Assistant Professor Sarunya Maneeroj, Ph.D. for suggesting and helping me understand the process of research, checking and correcting the thesis.

Secondly, I would like to thank all members of Assistant Professor Sarunya's advisees for giving and suggesting good solution and new prespective to me.

Thirdly, I would like to thank program chair, Associate Professor Preaphon Sophatsathit, Ph.D. and external examiner, Assistant Professor Saichon Jaiyen, Ph.D. for their valuable suggestions and comments for my thesis.

Fourthly, I am most grateful to my parents who have always supported and encouraged me in everything.

Finally, I am very thankful to everyone who has mentioned here and not mentioned above to helping and advising me for this thesis successful.

Weraphat Nimchaiyanan

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

In this chapter, we start to describe the background of interesting, problem and motivation in the first Section 1.1. Next, Section 1.2 shows the objectives. Then, Section 1.3 presents the scope and constraints of the thesis, followed by the expected outcomes in Section 1.4.

1.1 Background and Importance

A recommender system is a system that helps users find items by discovering patterns in a dataset and selecting the relevant items from patterns to recommend. For example, recommender system which uses ratings those predicts the user's ratings of each item that the users never rate before and provides the items that they would rate highly. Nowadays, recommender system is popular and applied in many areas. It is commonly used to generate playlists for music and videos in Netflix, YouTube and Spotify. Moreover, the most popular websites like Facebook, Instagram and Twitter also use recommender system to recommend contents which match the user preferences to others. Thus, recommender systems play an important role in helping users find products and contents that they want without having to spend all their time digging through things they would not like. Content-based filtering and collaborative filtering are the two main techniques in recommender systems.

Figure 1. An example basic Content-based filtering concept

Content-based filtering (CBF) is one of the techniques in recommender system. This technique recommends items that are similar to items that user used to previously rate. For example, in Figure1, the system recommends fruits to the target user by using CBF technique. She likes grapes. Strawberry is also recommended to the target user because grapes and strawberry have same characteristics which are sour taste, sweet taste and juicy. However, this technique has some drawbacks for recommendation. Firstly, the recommendation is not accurate and precise if the input data are not providing enough information to suggest the items precisely. Secondly, the results of recommendation from this technique may be over-specialization. It provides a limited degree of novelty because it perfectly has to match the features of profile and items. Thus, to suggestions from the results does not attract user's attention. Lastly, the recommendation to new users is in many case not provided correctly because they do not have enough information to build the user profile for perform the matching with the dataset.

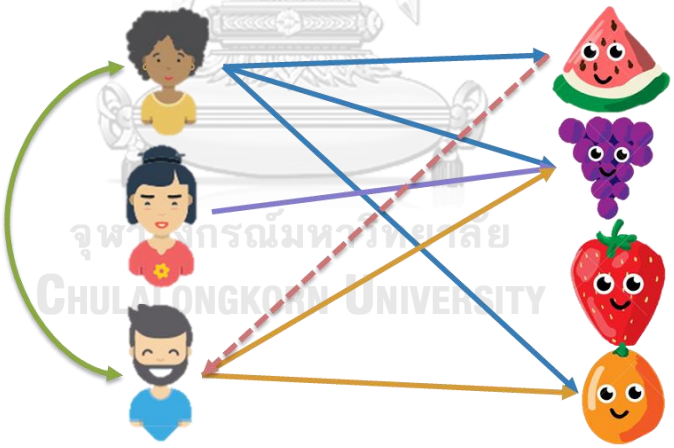


Figure 2. An example basic Collaborative filtering concept

On the other hand, Collaborative filtering (CF) is a technique to identify similar users who have similar preference to the target user. This group of similar users are called neighbors. Collaborative filtering will give items for recommendation based on the preference of neighbors. For example, in Figure 2, the system recommends fruits to the target user by using CF technique. He used to eat oranges and grapes. Then, the

system finds the neighbors of target users who have eaten oranges and grapes as same as the target user but recommends watermelons which the neighbors used to eat but the target user has never tasted before to him. CF technique has some several problems. First, this technique gets cold start problem which a new item needs to be rated by various enough number of users before it could be recommended while CBF does not has this limitation. Second, the problem of CF is grey sheep which users who are not consistent with their like, so CF recommendation is not reliable. To overcome the above problems from both techniques, the hybrid system is introduced from many researches. This technique and rating outcome are widely used in many researches for finding items or contents to target users. However, recommender system that only use rating is not enough because data of user are very sparse, and do not express user behavior in all situation.

The relevant contextual information does matter and becomes important for recommender system [1]. For example, the intent of a purchase made by a customer is used as contextual information. More specifically, the same user may buy different items for different reasons and situations: a book for improving her work skill, an accessory for a gift or an electronic device for her hobby. This example shows that the item that user's intention is considered by context information. This contextual information affects the user decision [2] because user preference changes all the time in accordance with specific situations. To deal with different intention of a recommendation, the profile of a user and models predicting recommendation behavior are built for all contexts. Thus, contextual information of a customer is useful because the results from predictive models are better than the traditional predictive models [3].

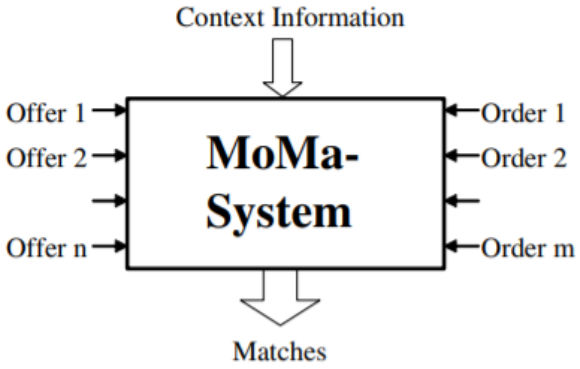


Figure 3. The principle of MoMa system [4]

Figure 4. A screenshot of MoMa-system on Symbian OS [4]

Context aware is used to consider for better understanding of user behavior and providing satisfied conditions to target users [5]. Bulander et al. [4] offered recommendations using a context aware for finding specific location called the MoMa-system. The basic principle of the MoMa-system is shown in Figure 3 and Figure 4 show in screenshot of MoMa-system at the end users. The target users create so called orders according to a given catalogue. This catalogue recommendation is a hierarchical ordered set of possible products which are described by appropriate attributes. On the uppermost level, it has “gastronomy” which have divided categories like “pubs”, “restaurants” or “catering services.” Each category is specified by certain attributes like” price level” and “style.” As above, it is put into system for creating an order or recommendation list. Then, the recommender system automatically fills in context

and profile parameters where appropriate with target user for more specifically example. After generating list of places, the recommender system tries to filter by context and provides the orders of the recommendation. The system would automatically add the appropriate physical context and profile parameters, for example, “location” and “weather.” The recommender system recommends the place closing to current location of the target user. It is raining, the outdoor location should not be recommended. This example uses location and weather as context for helping recommendation meet criteria of user attention.

Figure 5. An example basic Collaborative filtering concept

Currently, most researches show that the contextual information is used statically and the dataset including context information has not been changed since data has been created. However, user’s behavior can change all the time. Practically, context user that is used for prediction should be changed depending on situation [6]. This is called context driven. For instance, the price of the hotel room is dynamic depend on time context. More specifically, price in low season is cheaper than high season so this context is not stable. Thus, if the users are aware of prices, the system will add prices in each period of time to recommend hotels which meet their user attention. Another example, as shown in Figure 5, users do not have the same behavior in each season. They may like to reserve a hotel which is close to the beach in summer. On the other hand, they may like to stay in a luxury hotel with full facility in winter. If

these contexts that have been changed are provided in the predictive model, it will help meet their user attention.

The recommender system has many domains such as hotel recommendation, movie recommendation, tag recommendation, android application recommendation and TV program recommendation. These domains have high relation with context. The hotel recommendation is one of the domains which is interesting because this domain has high relation with context from reviews or comments. They in turn have rating which widely use in hotel recommendation. For instance, we tend to choose a hotel in different types or area depended on our proposal. There are also some recommended researches integrating Latent Dirichlet Allocation (LDA) into hotel recommendation for extracting values from reviews or comments without using user preference. In the hotel recommendation, reviews of hotel can be determined as context. Moreover, there are papers that use either CBF integrated with LDA or CF integrated with LDA. There is no paper that uses two main techniques and integrated LDA into their hotel recommendation. Thus, we propose a new method which using both Content-based filtering technique and Collaborative filtering technique and integrating context-driven by using LDA.

1.2 Objectives

1. A hotel hybrid recommendation method based on context-driven using LDA to improve recommendation list from existing method.
2. integrate context into recommender system for studying the effect of the context-driven to hotel recommendation.

1.3 Scope of thesis and constraints

This research uses the TripAdvisor in JSON form. Coverage will include the followings:

1. There are 878,561 users' reviews from 4,333 hotels
2. There are ratings (1-5) from 3,084 users on 4,333 hotels
3. The hotel details comprise of address of hotel, details, hotel class, hotelID, name, phone, regionID, type, url
4. The users comprise of author (assume that username is each user), date, date_stayed, hotel_id, num_helpful_votes, offering_id, ratings, text(comment), post_title, via_mobile

1.4 Expected Outcome

The proposed method will provide high accuracy and better results then those using CBF integrating with LDA and CF integration with LDA. Moreover, better performance than traditional methods can be achieved when context change is taken into account.

1.5 Thesis structure

This thesis consists of 4 main chapters, which Chapter 1 provides the introduction. Chapter 2 contains the principle knowledge background and literature reviews. Chapter 3 explains the methodology and analysis of data. Chapter 4 discusses the experimental results and some final thoughts in the conclusion.

CHAPTER 2

THEORETICAL BACKGROUNDS AND RELATED WORK

There are many principles and techniques which are used in recommender system to improve the performance. These techniques are used to understand user intention and behavior in recommendation each technique will be described. Section 2.1 focuses on the meaning of recommender system and techniques which are used in this work. Section 2.2 describe related works which are used in this work. Followed by current situation in hotel recommendation in Section 2.3

2.1 Principles of recommendation and techniques

2.1.1 RECOMMENDER SYSTEM

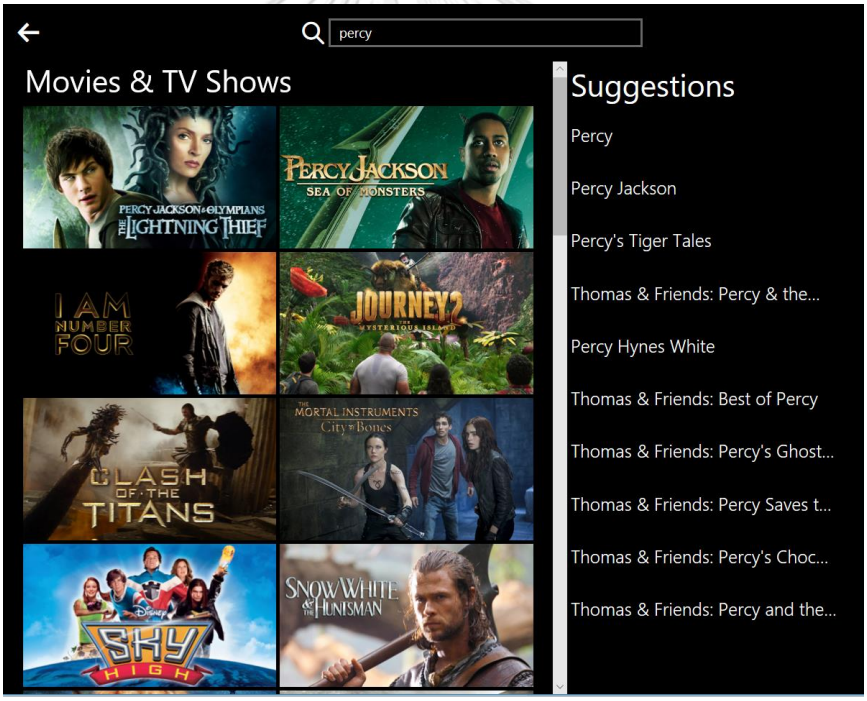


Figure 6. The result of searching some items on NETFLIX

Nowadays, internet becomes an important means for users in many facets depending on individual users. Users have come across a recommender system in some way because there are a lot of information in the internet. They do not want to spend times finding the target information by each transaction. For example, your friend recommends a new movie to you, but you have never seen it. Then, you visit

your favorite online movie store. After typing in the title of the movie, it appears as just one of the results listed. In the web page, it does not show only the result that you already searched, but also shows another section in the webpage that called “Customers Who Brought This Item Also Bought”, a list is shown of additional movies that you may be interested in. From Figure 6, when users type some words in the search field and the results that are provided to users do not show only items that exactly match with keywords, but they also showed items that user may intent with them. In addition, if you are usually use the same online movie store, such a personalized list of recommendation will appear automatically when you enter the store. The software system determines which movies should be shown to a visitor is a recommender system. As show in Figure 7, when the users login to the website, suggested movies are showed in recommendation tab.

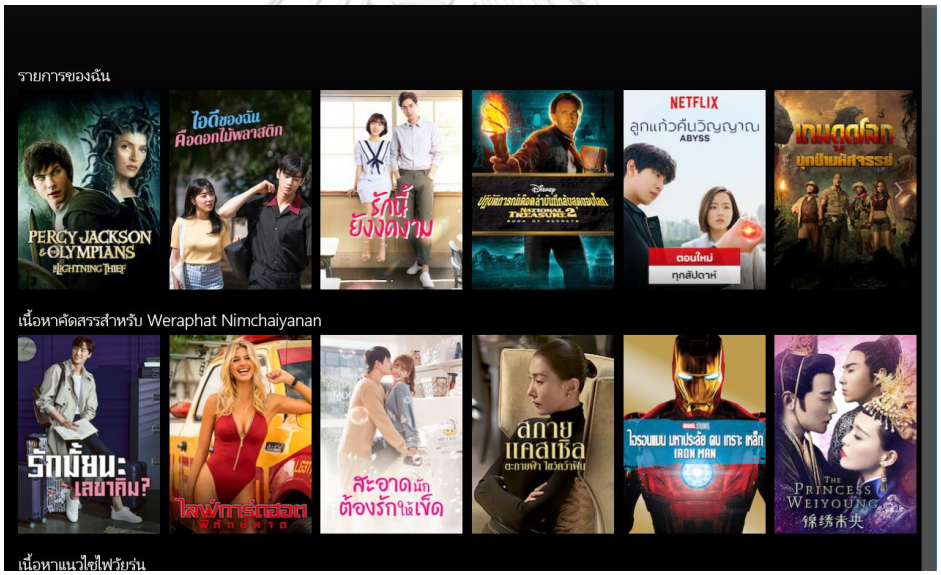


Figure 7. The recommendation items after logging on

From the above example, it is useful to focus on systems perspective software. The main logic that behind the recommender system is personalized recommendations which means every visitor sees individual list of recommendation depending on their interest. However, there are some online shops recommend you by using their top seller items or their most favorite read articles. In this case, they

interpret the information as impersonal buying or reading recommendation. Although the top seller suits many users, some users may not appreciate them as well. For instance, Avatar is a popular movie but there will be some people who do not like to see despite it is the highest-grossing films in 2009. From this case, recommending by using top items is not very helpful for them. Thus, the system that is generated by personalized recommendation can be very effective and express interest of individual.

The Provision of personalized recommendations requires that the system must know some information from every user for using recommended. Every recommender system must collect and maintain a users' information or call user profile or user model. For instance, in our online movie store example, the system collects user's preferences by recording which movies that a visitor has seen in the past for using prediction which other movies might be of interest.

Every recommender system refers user profile as a core of it but the way in which the user's information is received depends on each recommendation technique. User preferences can be acquired implicitly by observing or gathering from user behavior or explicitly by asking users about their preferences. And the basic idea of recommender systems are content-based technique and collaborative filtering technique.

2.1.2 CONTENT-BASED FILTERING TECHNIQUE

The main idea of content-based filtering (CBF) approaches is to exploit information about a user profile which are users' interest to items in the past for predicting the same items. These items have characteristics matching with user profile and recommend them to users. In some situation, you may approach with content-based filtering. For example, when your friend asks you to recommend new books to him, some questions which you may ask him are the kinds of books he used to read. From there, you can think of a few books that are similar to the things he has liked in the past and give some names which have matching characteristics with your friends.

Table 1. Characteristics of books in databased

Title	Genre	Author	price	keywords
The Lion King (Little Golden Book)	Children	Justine Korman	$2.99	cartoon, movie, animal
The Lace Reader	Fiction, Mystery	Brunonia	$29.90	fiction, detective historical
The Lightning Thief (Percy Jackson and the Olympians)	Fantasy, Adventure	Rick Riordan	$5.99	Olympus, supernatural, movie

Table 2. The preference Profile

Title	Genre	Author	price	keywords
The Little Mermaid	Children, Fantasy	Hans Christian Andersen	$5.59	cartoon, beach, movie

For example, recommendation by using content-based filtering technique normally uses characteristics of items. The information which is used for recommendation is provided from an explicit list of features for each item. The table 1 describes characteristics of books in the database, including title, author, genre, price and keywords. The target user preferences have the exact same dimensions as shown in Table 2. This means when he has selected items, those are added into user profile in the database to collect as preferences. The concepts of content-based filtering technique are to find items which target users have not seen before and evaluate the similarity with the items that users like them. The similarity can be measured in different ways. The easy way to find the recommending items by matching some target user preferences with characteristics of items is price. If the target users' preferences are mapped by price, the system will recommend The Lightning Thief (Percy Jackson

and the Olympians) to the target users because the price is the closest to each other. Another example, we find the similarity by using keywords which rely on Dice coefficient as shown in Equation 1: Book b_i and Book b_j are described by keywords. The similarity of this case is between 0 to 1. It is suitable for muti-valued characteristics. From the Table 1 and Table 2, If we calculate them by this equation, the similarity of target user with each book is: with The Lion King (Little Golden Book) = 0.67, The Lace Reader = 0.0 and The Lightning Thief (Percy Jackson and the Olympians) = 0.33. So, the system provides The Lion King (Little Golden Book) book to the target users.

$$\frac{2\times|keywords(b_i)|\cap|keywords(b_j)|}{|keywords(b_i)|+|keywords(b_j)|} \quad (1)$$

This technique is behind search engines of some reputation websites such as Netflix and Pandora's recommendation engines because this technique has some advantages. The first advantage is that new items can be suggested before being rated until reached a substantial number of users because this approach rely on characteristics of items and user preferences. It does not require the existing of large users' community or rating history. Thus, recommendation lists can be created although there is only one user in the system. When content-based is compared with collaborative technique, the process has a black box–the algorithm to calculate neighbors' references for recommendation. This means collaborative technique does not express characteristics of items directly. On the other hand, the recommendation list from content-based filtering is transparent because it is not depending on other users to recommend.

However, content-based filtering also has some limitations. There are 3 main problems which are shallow content analysis, overspecialization and acquiring rating. Firstly, for shallow content analysis, items which are recommended by capturing from quality or characteristics of items alone may not be enough. Information of items is more contained than only rating or characteristics of items. It contains many elements such as comment to user to the items, images and videos so these one should be

concerned for recommendation. Secondly, for overspecialization, the recommendation list which provide from this technique can provide only items that are similar to the ones the user has already rated. This can be not suitable for some users who want to divert type of recommendation for example, recommendation about news, the system will provide the articles which cover the same articles that users have ever seen which lead to the undesirable to users. Lastly, for acquiring rating, although content-based filtering does not require large users' community for recommendation, it have to require some rating form user or some explicit statement such as 'like' and 'dislike' to generate the recommendation list to user.

2.1.3 COLLABORATIVE FILTERING TECHNIQUE

The main idea of collaborative filtering (CF) approach is to take advantage of users' historical preference on a set of items for predicting which items match with the users the most. This technique does not need to characteristics of items to interpreted recommendation list. For example, you have close friends who like the same kind of food as you. When you go with him to dinning and he orders noodle, you may order noodle as well because both of you have the same past eating behavior.

Table 3. The user-item rating matrix

	Item1	Item2	Item3	Item4	Item5
Target users	5	3	4	4	?
User1	3	1	2	3	3
User2	1	5	5	2	1
User3	4	3	4	3	5

For Example, Recommendation by using collaborative filtering identifies other users that have similar preferences to the target user in the past. It predicts rating to items that the target user has not seen them by using neighbors. Table 3 shows user-item rating matrix. Rating in this table is rated on 1 to 5 scale. The score is high which means users love that item. From this table, our proposed is that we want to find users who have the same preferences with the target users and use the rating of this group

for Item5 to predict rating of the target users for Item5. The measurement uses in recommender system to find the similarity is Pearson's correlation coefficient. The similarity *sim(a,b)* of users *a* and *b*, given the rating matrix *R*, is show in Equation 2

$$sim(a,b) = \frac{\sum_{i \in I}(r_{a,i} - \overline{r_a})(r_{b,i} - \bar{r}_b)}{\sqrt{\sum_{i \in I}(r_{a,i} - \bar{r}_a)^2}\sqrt{\sum_{i \in I}(r_{b,i} - \overline{r_b})^2}} \quad (2)$$

We introduce the symbols and parameters of Equation 2. We use *I* denote the set of items; *R* is user-items rating matrix, $r_{a,b}$, $r_{b,i}$ denoting rating of user a or user b. For item *i* and $\bar{r}_a, \overline{r_b}$ denotes average rating of user *a* or user *b*. The Pearson correlation efficient takes value from -1 to +1. The number -1 is strong negative correlation which means the relation between both users is negative and there is opposite relationship. On the other hand, the number +1 is strong positive correlation which means the relationship between both users is positive and there is the same direction of relationship. After the comparing target user with individual users in the dataset from Table 3, The similarity of target user to *user1* is 0.85, the similarity of the target user to *user2* is -0.79 and the similarity of target user to *user3* is 0.70. user*1* and *user3* are similar to the target user's preference in the past. After, we get user*1* and *user3* as neighbors to predict *Item5* of the target users, they are computed to find predicted ratings of target user as Equation 3

$$pred(a,i) = \bar{r}_a + \frac{\sum_{n \in N} sim(a,b) \times (r_{b,i} - \bar{r}_b)}{\sum_{n \in N} sim(a,b)} \quad (3)$$

We introduce the symbols and parameter of Equation 3. We use $pred(a,i)$ to denote predicted rating provide to user *a* for the item. In this case, we predict rating of *Item5* to target user by using neighbors which are $user_1$ and $user_3$. The prediction after calculation following Equation 3 is 4.87. Currently, we can compute predicted rating for target user for all items that she has never seen. However, In the real recommender system, users, items and ratings in databased have billion records so the recommender system must be complexity more than this example.

Due to collaborative filtering relying on the behavior of users, that making this technique has some strong point over content-based filtering. The recommendation list is very effective without implement additional development work when there is a large user databased. It also provides diverse recommendation list because collaborative filtering is based on relation between users and it can make connections between seemingly disparate items for instant, in the databased, you and your friend like fishing and we have same many type of fish in databased but your friend has data about fishing rods. Therefore, the recommend list is provided about fishing rods to you. It does not only provide.

Due to high demanding of information for recommending, that make it has some drawback. Firstly, the cold start problem occurs from new users, new items, and new systems, where recommendation is not possible as the information about the user is not enough or rating for the product is limited due to new items or new systems so collaborative filtering is unable to make accuracy recommendations. Secondly, scalability occurs when this technique has to match target users with other users that have same behavior so the number of users that we have to use are the main factor for predicting. Each user has to select items enough to cope up with the increasing number of items and make the efficiency of it still be acceptable. Lastly, sparsity of data is one of the crucial factors frequently encounter by this technique. It exists as the user does not rate most of the items and the ratings of the items remain spars.

2.1.4 CONTEXT-AWARE RECOMMENDER SYSTEM

The traditional recommender systems which apply content-based filtering and collaborative filtering, use simple user models and most of them only using rating for recommendation. It does not enough for recommending because it cannot express user interested in all situation. For example, user-based collaborative filtering model sees the vector of item ratings in each user as users' preferences. Every user has their own preferences in the dataset, and they are used to generate recommendations or make predictions. Thus, this model has not considered about contextual information which may has effect for their decision. The fact that users interact with the specific contextual situation of the user so users' preferences within one contextual information may be different from those in another contextual information and the

recommender system that it considers by contextual information can call “context-aware recommender system”

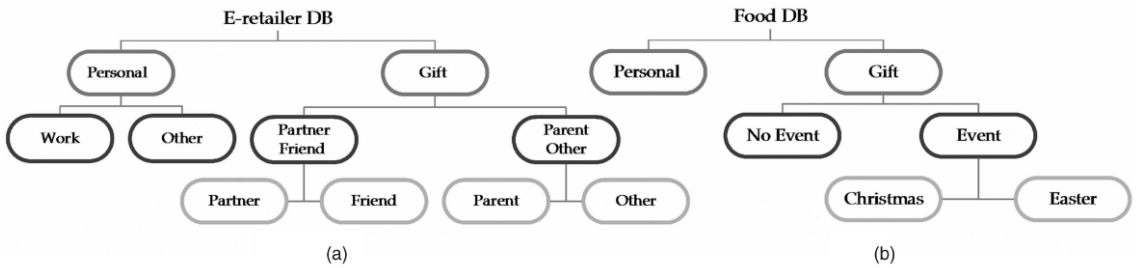


Figure 8. The contextual information hierarchical structure

Contextual Information such as time, place and intention of user is information that gives context to a person, entity or event. It is applied to improve performances for recommendation and to help us understand users’ behavior because some domains in recommender system are not sufficient to consider only users and items such as hotel recommendation. As for user’s contextual information in the hotel recommendation databased include many information such as users’ reviews, location, facility and accommodation. it is also important to incorporate the contextual information into the recommendation process for recommending items to users which match with context of users. For example, a hotel recommender system would provide hotels in specific type of hotel depend on user’s intention because choosing hotel to stay with friends may be very different from the one staying with family. Moreover, contextual information is popular in e-commerce because this domain also high contextual information. For example, Cosimo Palmisano et al. [2] presented using context to improve predictive modeling of customers in personalization applications. They integrated contextual information into predictive model. As Figure 8, it shows hierarchy for dealing with different the contextual attribute of specific intent of a purchasing transaction in this paper. They used the intent of a purchase made by a customer in an e-commerce application as contextual information. Different purchasing intents of users might lead to different types of behavior. For example, the same customer shopped from the same online account different products for different

reasons: a book for her personal work skills, a book as a gift, or a fashion for her hobby. They built a separate profile of a user for each purchasing context to deal with different purchasing intentions and these separate profiles were used for building separate models predicting user's behavior in specific contexts and for specific segments of customers as show in Figure 8. They created predictive model for hierarchical structure in individual profile. Their proposed model is useful, because it resulted in better predictive models across different e-commerce applications.

To improve recommender system, it is not only used rating for recommendation, but contextual information also consider for recommendation to express users' preferences that hidden in users. this thesis also concerns about context within users' preferences via users' reviews to the hotel to use for improvement recommendation list.

2.1.5 CONTEXT-DRIVEN RECOMMENDER SYSTEM

In the past, context in context-aware recommender systems is used to generated recommendation list statically that users' goals do not change since user's data has been created in the system so the set of items to be recommended remains relatively static. Context-driven is improved to revise this problem. For example, they introduce to prioritize present preferences over past preferences, they provide the recommendation list including only currently available items or they adjust weights to recommendation list depending on seasonality or trends [2].

Context-driven recommender system is considered to overcome using statically data within recommendation. It concerns about current users' preferences or preferences changes because in the real world, users' preferences are not consistent. In the contextual information, it focuses on item users' preferences which since data is created, for specifically example, users like to watch movie when these preferences were capture in the recommendation, it has been not change which means every time in recommendation list is provide to him. He will always see action movie. It is not good for user and the system because currently, he may not like action movie. Thus, context-driven help improvement. Users' preferences or context are split into fragments of different dynamics, each one reflecting her different intents and situations or times. By the mean of this break-down, context-driven algorithms are able

to model a more fine-grained similarity: the similarity which take places from comparing contexts in the same situations or times. For instant, from above example, the recommendation list can be improvement by finding the present context of user which movie type he likes currently. Then, provide recommendation which has similarity with his currently preferences.

In movie Domain, Thitiporn Neammanee, et al. [7] presented time-aware recommendation based on user preference driven. their paper proposes consider time to find the preference change in the rating timeline. In their work, they captured user currently preference by time. They started to find period of data in the past which similar preference to current period's preference of target user. Then, they found the neighbors of the target user who has the highest degree of membership in the same cluster with the target user. Finally, they predicted the rating for the target item by averaging weight on neighbor rating. The advantage of their paper is that they only used some fragment of contextual information instead of using all context in dataset. And the result provided outstanding accuracy and coverage when compared with recommendation which only focused on contextual information.

The power of context-driven can provide relevant context with target users better than only use context constantly so it is very interest to integrate context-driven into our proposed method to improve algorithm instead of applying directly contextual information.

2.1.6 LATENT DIRICHLET ALLOCATION

As knowing that contextual information has an impact for recommendations, some of contextual information cannot be input of the recommendations directly such as reviews or comments so it has to be changed to be scalable before putting it in predictive model. Text mining is a method for extracting contextual information through the identification and exploration of large amounts of text. It applies techniques such as categorization, entity extraction, sentiment analysis and natural language processing.

Probabilistic topic models are algorithms for discovering the latent semantic structures from extensive documents. It uses probability to describe hidden structure inside the documents. Latent Dirichlet Allocation (LDA) is a generative probabilistic

model. It is one of the probabilistic topics models that is popular in natural language processing (NLP) field. It was proposed by David Blei [8]. The basic idea of LDA is that documents are represented as random mixtures over latent topics which are hidden inside the documents, each topic is characterized by distribution over words. For example, topics might be science, biology, and physiology. Thus, for LDA, documents and words are observed variables, and latent topics are calculated by using distribution of words which is conditioned on the document. LDA is considered unsupervised learning algorithm because the latent topics output is received by taking only a word-document co-occurrence matrix. Although LDA works with unlabeled input, it can provide desired result in terms of human interpretation.

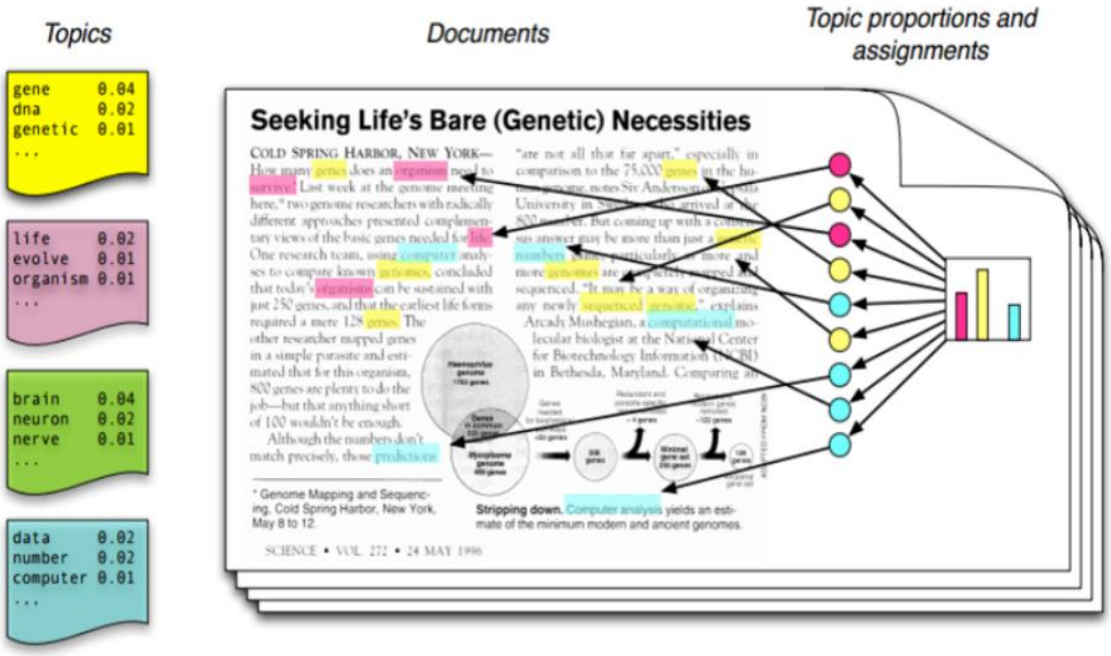


Figure 9. The LDA intuition

Figure 9 shows an overview of LDA intuition, they assume that the author who writes the document has certain topics in his mind. As for writing about a topic then means to pick a word with a certain probability from a box of words of that topic. For example, imagine that there are many documents about mathematics and there are three chosen latent topics. LDA assumes that words about number, integer, or float are grouped into the first latent topic with high probability. Words about rectangle,

triangle, or pentagon are grouped into the second topic with high probability, and words about degree, angle, or radius are grouped into the third topic with high probability. Furthermore, each word can be grouped into more than one latent topic by probability. For example, Triangle may be categorized into the first topic and the second one. However, the probability of words about triangle in each topic is not equal. For each document in mathematics, it consists of different mixture of this latent topic. Regarding to knowledge, how the latent topics are categorized by words and how the topics are distributed across documents will be information for new document generation. Thus, LDA is also considered into the generative model because after running the model by parameters, a new document which is a probability distribution of documents and words over the latent topics is generated.

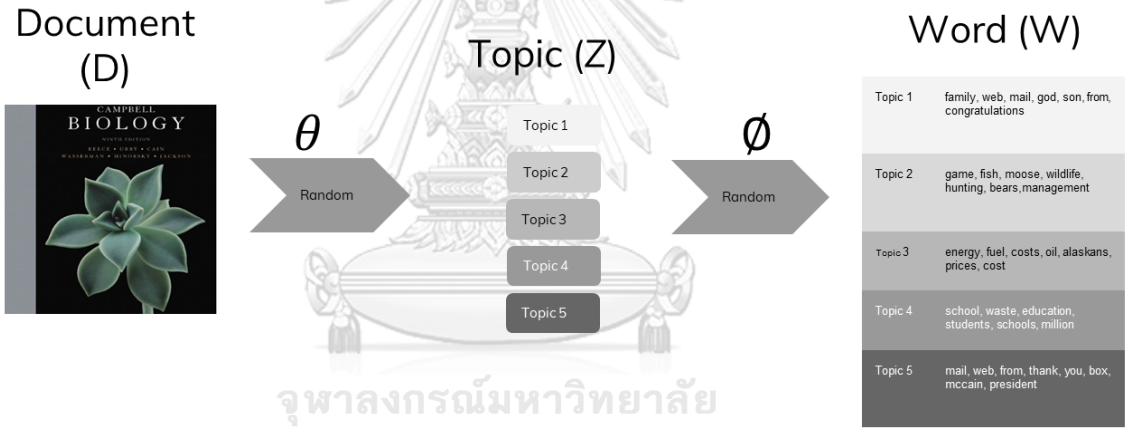


Figure 10. The LDA notation in LDA model

From a mathematical view, we can show LDA in a generative process of the probability. We show some notation in Figure 10 and denote each notation and its meaning as in Table 4 before we explain about it. Remark that when topics are referred, it is referred to latent topics in LDA.

Table 4. Meanings of LDA notation

Notation	Meaning
N	Number of words in the corpus
M	Number of documents
K	Number of latent topics
w	Set of words which have N vocabulary
D	Set of documents which have M documents
$\emptyset$	Topic-word distribution which is multinomial distribution
θ	Document-topic distribution which is multinomial distribution
Z_n	Topic assignment of nth words
β	The prior distribution of ϕ which is Dirichlet distribution
α	The prior distribution of θ which is Dirichlet distribution

In generative process, LDA is described by assuming that the latent topics are pre-defined before any data is generated. For each document, each word can be generated without considering grammar. The generative process steps are ordered by the following. Firstly, each topic is randomly selected a topic distribution over words as written into mathematical term below.

$$For\ k = 1 \ldots K, \emptyset_k \sim DIR(\beta)$$

Secondly, each document is randomly selected a topic distribution over documents as written into mathematical term below.

$$For\ d = 1 \ldots D, \theta_d \sim DIR(\alpha)$$

Lastly, each word index is considered in 2 terms. In the first term, it randomly selects a topic from document-topic distribution. After that, it assigns a topic to each word index as written into mathematical term below.

$$For\ each\ w_d \in\ d, z_{w_d} \sim Multi(\theta_d)$$

In the second term, it randomly selects a word from the drawn topic in the first term from a topic-word distribution as written into mathematical term below.

$$For\ each\ w_d \in\ d, w_d \sim Multi(\emptyset_{z_{w_d}})$$

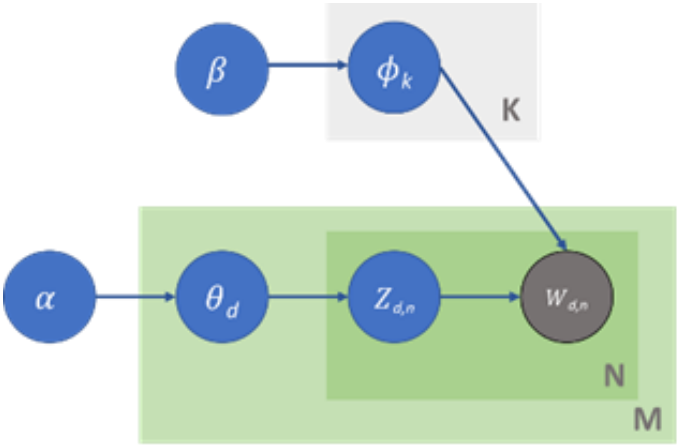


Figure 11. The graphical model of LDA

From the generative process, it is represented as a probabilistic graphical model as in Figure. 11. The probabilistic graphical model is a modeling language which helps to understand and represent relationship of the probability distribution. Each node in the graphical model represents random variables. The topic node, document-topic distribution node, topic-word distribution node, and two prior distribution are blue-shaded because they are unobserved variables that are needed to suggest. The word node is gray-shaded because it is only one observed variable in the graphical model. The rectangle bounding in the graphical model represents each iteration. The N plate denotes iteration over a number of words. The M plate denotes iteration over a number of documents, and the K plate denotes iteration over a number of topics. The arrows represent relation between random variables. For example, words are drawn from the topic assignment. Thus, there is a one-direction arrow from the topic assignment node to the word node.

From the generative process and the graphical model, LDA can also be represented as the joint probability distribution of the hidden variables and observed variables as in Equation 4.

$$
\begin{aligned}
& P(Z, W, \theta, \beta \mid \alpha, \eta) \\
& = P(\beta \mid \eta)\, P(\theta \mid \alpha)\, P(Z \mid \theta)\, P(W \mid Z, \beta) \qquad (4) \\
& = \prod_{k=1}^{k} P(\beta \mid \eta_k) \left[\prod_{d=1}^{M} P(\theta_d \mid \alpha) \left(\prod_{w=1}^{N} P\left(Z_{d,w} \mid \theta_d\right) P\left(W_{d,w} \mid \beta_{1:k},\, Z_{d,w}\right) \right) \right]
\end{aligned}
$$

The probability distribution joint also shows dependencies of each random variables. For example, the observed word $W_{d,w}$ is generated from topic assignment $Z_{d,w}$ and topic-word distribution. For the goal of LDA, it is to learn hidden structures from observed data. Thus, this problem is turned into reversing the generative process and learning the posterior distribution of the latent variables giving the observed data. The posterior distribution of LDA can be shown in Equation 5.

$$P(Z,\beta,\theta) = \frac{P(Z,W,\beta,\theta|\ \theta,\eta)}{P(W|\ \alpha,\eta)} \quad (5)$$

2.2 Related work

The LDA topic model is used in recommender systems in many domains for helping content-based filtering and collaborative filtering technique to recommend in many researches as the following example below.

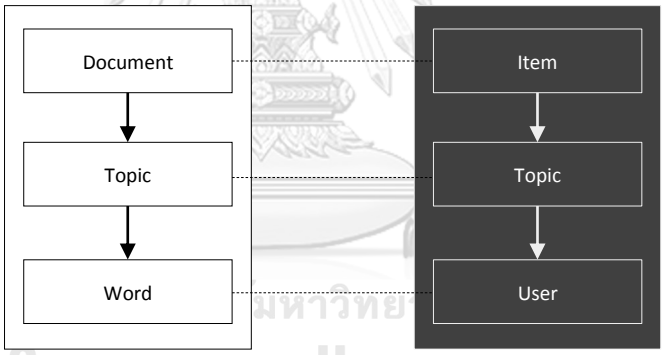


Figure 12. The analogy between users and items into LDA

Liu Na, et al. [9] presented the improved collaborative filtering algorithm using topic model. They wanted to recommend item to target users. LDA is applied to find the hidden topic in user-item matrix. The users acted as documents and items acted as words in Figure 12. After applying LDA, they got item-topic distribution (θ) and user-topic distribution. It was determined by rating under users and item-topic distribution by using Equation 6.

$$\theta_{u_p}^{t_q} = \sum_{k \in M} r_{i_k} \times \theta_{i_k}^{t_q} \times \varphi_q \quad (6)$$

After that, they found the similarity of LDA between target users and users (neighbors) on user-topic matrix by using KL divergence as in Equation 7 for using a part of total similarity in Equation 8 of the author and they computed the similarity to find neighbors of target users as Equation 8.

$$sim_{i,j}^{\mathrm{LDA}} = exp^{-\left(\sum_{k\epsilon M} \ln\left(\frac{\theta_i}{\theta_k}\right)\theta_i + \sum_{k\epsilon M} \ln\left(\frac{\theta_j}{\theta_k}\right)\theta_j\right)} \quad (7)$$

$$sim_{i,j} = \lambda\left(\frac{1}{3}\left(sim_{i,j}^{c} + sim_{i,j}^{p} + sim_{i,j}^{ac} +\right)\right) + (1-\lambda)sim_{i,j}^{\mathrm{LDA}} \quad (8)$$

A Recommendation list was provided to target users by predicted ratings which were calculated from neighbors of target users as Equation 9.

$$\hat{r}_{u,i} = \sum_{j\epsilon M_i} sim_{i,j} * r_{u,j} / \sum_{j\epsilon M_i} |sim_{i,j}| \quad (9)$$

Their work also has some shortcomings: They did not concern about latent context such as hotel reviews. The hotels which got high ratings did not mean that the target users like them [10] so they should consider another factor to improve their recommendation list.

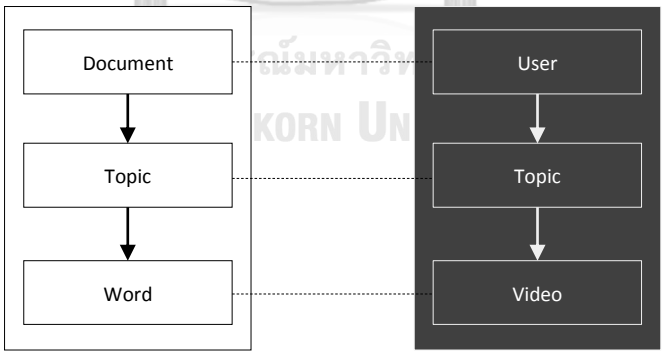


Figure 13. The analogy between users and videos into LDA

Jie Zhang, et al. [11] presented IPTV program recommendation by using an implicit feedback integrated LDA topic model. In their work, Users acted as documents and videos acted as words in LDA as shown in Figure 13.

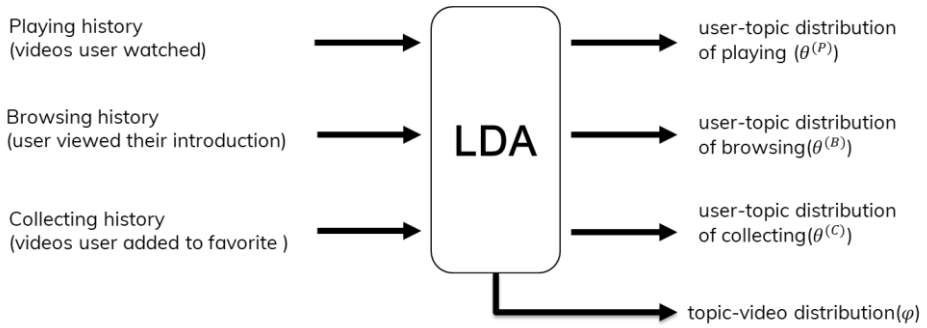


Figure 14. The graphical model of the multinomial distribution

They divided a set of topics of the multinomial distribution of each user ($\boldsymbol{\theta}$) into 3 parts which are playing history (videos user watched), browsing history (user viewed their introduction but not watched) and collecting history (videos that user added to favorite but not watched) as shown in Figure 14. The real $\boldsymbol{\theta}$ was obtained by combining each $\boldsymbol{\theta}$ and the regression coefficients $\boldsymbol{\omega}$ as shown in Equation 10.

$$\theta = \omega_1\theta^{(P)} + \omega_2\theta^{(B)} + \omega_3\theta^{(C)} \quad (10)$$

They also got videos distribution for Topics ($\emptyset$) from LDA. After that, they used content-based filtering to multiply real $\boldsymbol{\theta}$ and $\emptyset$ to get rank score and the videos that have Top-N of high score will be recommended.

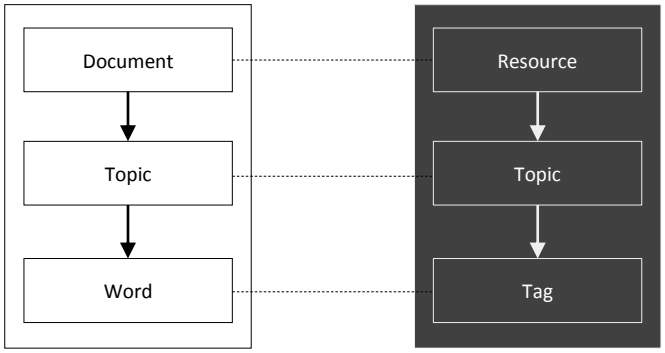


Figure 15. The analogy between Resource and Tag into LDA

Ralf Krestel, et al. [12] presented LDA for tag recommendation in order to improve search. They recommended a set of tags for a target resource. The resource acted as document and the tag acted as words in Figure 15.

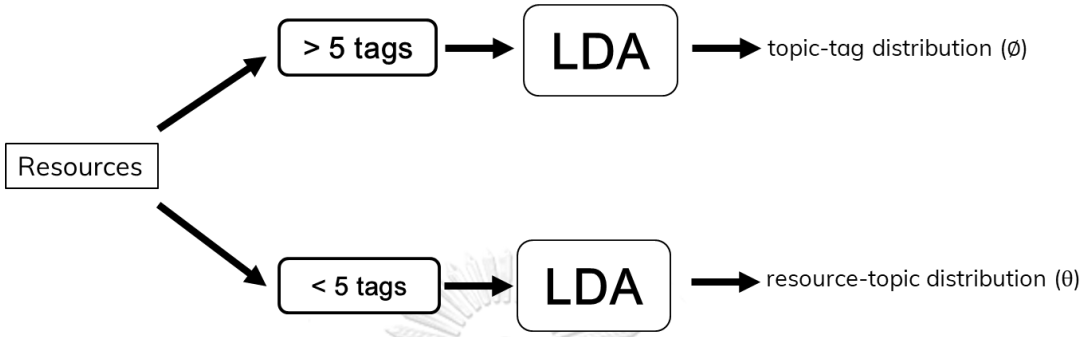


Figure 16. The structure of tag recommendation

In Figure 16, it shows the structure of their model for using recommendation tag to users. The resources that have more than five tags were used to build LDA model in order to obtain Topic-tag distribution (Ø) to show that which tags should be in which topics. For recommending sets of tags which have less than five tags used as a target resource, firstly, the latent topic distribution of the target resource is identified using LDA as being shown. After that, content-based filtering technique will be applied by multiplying θ of the target resources with the derived Topic-tag distribution (Ø). Finally, the tags that have multiplied score more than the threshold will be recommended for the target resource.

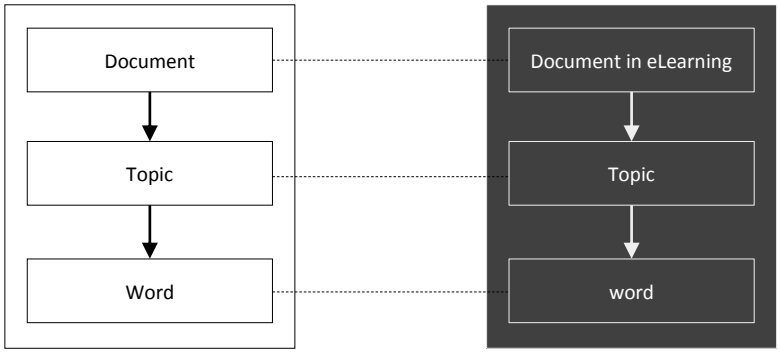


Figure 17. The analogy between document and word into LDA

Rohit Nagori, et al. [13] presented LDA-based integrated document recommendation model for e-learning system. This paper applied LDA as baseline for LDA principle as shown in Figure 17. All documents in e-learning system were used to build LDA model in order to get document-topics distribution for each document (θ) to show the ratio of all topics in each document. To recommend new documents for a target user, firstly, they find the document-topics distribution of documents (θ) that the target user has studied to represent his preference. Finally, a set of new documents were recommended by applying content-based filtering to finding predicted ratings. The documents found were those that had similarity of topics with target user's past documents as shown in the following Equation 11.

$$sim(p,q) = \frac{\theta_{[p]} \cdot \theta_{[q]}}{\|\theta_{[p]}\| \|\theta_{[q]}\|} \quad (11)$$

From the above equation, $sim(p,q)$ denotes similarity between document *p* and *q*. *p* and *q* refer to documents and $\theta_{[p]}$ and $\theta_{[q]}$ refer to document-topics distribution for *p* document and *q* document respectively. Then, the similarity was used to find predicted ratings of documents that target users never studied before as shown in Equation 12.

$$P_{ai} = \frac{\sum(r_k \cdot sim(p,q))}{\sum sim(p,q)} \quad (12)$$

From Equation 11, P_{ai} denotes the predicted rating of *i* item for *a* user. r_k refers to rating of *k* item that user used to rate. The recommendation list was created from the above equation and the recommendation list was sorted by predicted rating.

2.3 Current situation in hotel recommendation

Hotel recommendation is one of many domains which exploits context for recommendation because there is high contextual information of both users and hotels: users' information such as users' intent, destination and users' reviews, hotels': location, nearby places and hotel reviews. Most of hotel recommendations apply either content-based filtering technique or collaborative filtering technique to

recommend. Although, some researches apply both techniques but there is no research that applies both techniques with LDA. Moreover, hotel recommendation normally uses only rating for predicting model. Nobody concerns about using both ratings and users' reviews which have high contextual information and using context-driven to understand users' behavior in hotel recommendation. Thus, in this research, we focus on both traditional technique and ratings and reviews that extract value by using LDA and applying context driven to improve performance of recommender system.

CHAPTER 3

PROPOSED METHOD

In this chapter, the researchers' proposed methods and algorithm are described. As for recommending hotels, there is not method that uses hybrid between content-based and collaborative filtering by applying LDA to extract contextual information from users to provide recommendation. Moreover, there is no hotel recommendation that uses both ratings and reviews to be input in order to analyze the recommendation and consider context-driven from contextual-information for hotel reviews. Thus, our study proposes a hotel hybrid recommendation method based on context-driven using LDA. Figure 18 shows the overview of our proposed method. From the figure, we use users' ratings and reviews as input and it goes through many processes and applies reviews by LDA to find latent topic hidden in context and provide some context driven by time to extract current users' preferences and get recommendation lists to target. In this section, we divide our step into 5 steps; 3.1 Data preparation, 3.2 Finding missing rating, 3.3 Finding the similarity of users, 3.4 Incorporating context-driven to recommend hotel and 3.5 Creating recommendation list.

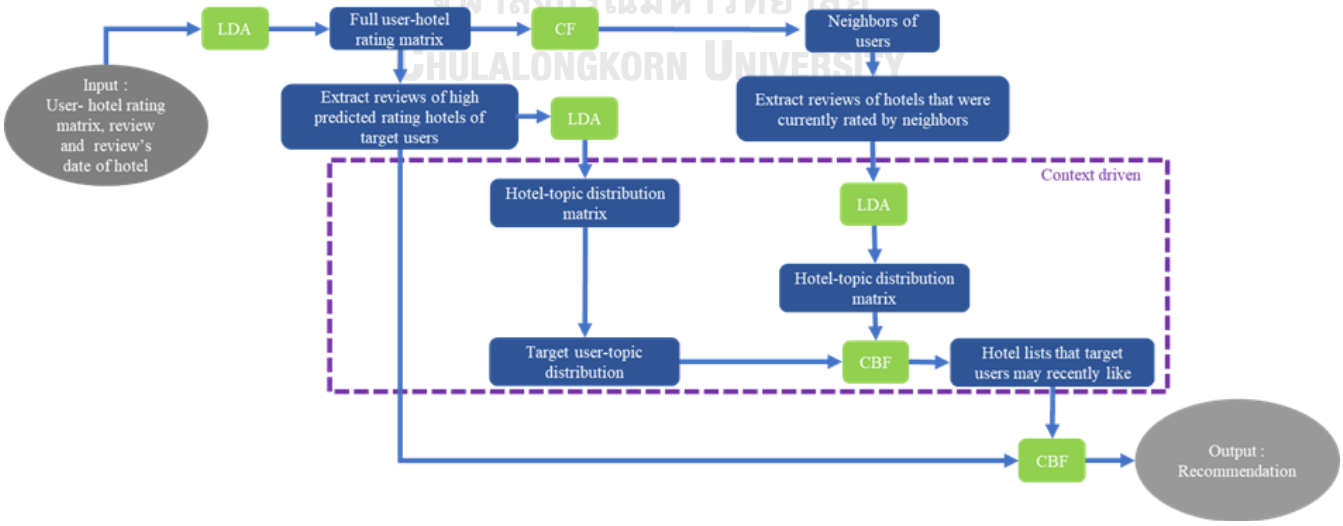


Figure 18. Our recommender system structure

3.1 Data preparation

Dataset is crawled from TripAdvisor by Myle Ott who is a research engineer in Facebook's AI Research group. In the dataset, there is a lot of information including rating of users that was provided to hotels, reviews of users that was provided to hotels, location of hotels, countries that the hotels are in, title of reviews and date of stay of users. All of raw data set is collected into JSON form as shown in Table 5 which shows the first fourth of raw dataset. In each record of dataset, there are 5 main parts of the information which are 1). rating which contains rating of service, rating of cleanliness, rating of overall, rating of value, rating of location, rating of sleep quality and rating of room; 2) the title of users' review; 3) content of users' review; 4) users' information which contains username, number of users' activity and Id of users; 5) general information which contains date stay, creating date and etc. Moreover, the raw data has another file which collect hotel information such as hotel name, location etc. but we discard all of this information because it does not represent users' preferences and it is not necessary for using our proposed method. However, our proposed methods only consider the overall of users' ratings that were provided to hotels, users' reviews that were provided to hotels which represented users' preferences and comment date used for considering context-driven.

Table 5. The raw dataset which crawled from TripAdvisor in JSON form

No.	Content
1	[“ratings": [“service": 5.0, "cleanliness": 5.0, "overall": 5.0, "value": 5.0, "location": 5.0, "sleep_quality": 5.0, "rooms": 5.0], "title": "\u201cTruly is \"Jewel of the Upper Wets Side\"\u201d", "text": "Stayed in a king suite for 11 nights and yes it cots us a bit but we were happy with the standard of room, the location and the friendliness of the staff. Our room was on the 20th floor overlooking Broadway and the madhouse of the Fairway Market. Room was quite with no noise evident from the hallway or adjoining rooms. It was great to be able to open windows when we craved fresh rather than heated air. The beds, including the fold out sofa bed, were comfortable and the rooms were cleaned well. Wi-fi access worked like a dream with only one connectivity issue on our first night and this was promptly responded to with a call from the service provider to ensure that all was well. The location close to the 72nd Street subway station is great and the complimentary umbrellas on the drizzly days were greatly appreciated. It is fabulous to have the kitchen with cooking facilities and the access to a whole range of fresh foods directly across the road at Fairway.\nThis is the second

	time that members of the party have stayed at the Beacon and it will certainly be our hotel of choice for future visits.", "authors": [“username": "Papa_Panda", "num_cities": 22, "num_helpful_votes": 12, "num_reviews": 29, "num_type_reviews": 24, "id": "8C0B42FF3C0FA366A21CFD785302A032", "location": "Gold Coast”], "date_stayed": "December 2012", "offering_id": 93338, "num_helpful_votes": 0, "date": "December 17, 2012", "id": 147643103, "via_mobile": false]
2	[“ratings": [“service": 5.0, "cleanliness": 5.0, "overall": 5.0, "value": 5.0, "location": 5.0, "sleep_quality": 5.0, "rooms": 5.0], "title": "\u201cMy home away from home!\u201d", "text": "On every visit to NYC, the Hotel Beacon is the place we love to stay. So conveniently located to Central Park, Lincoln Center and great local restaurants. The rooms are lovely - beds so comfortable, a great little kitchen and new wizz bang coffee maker. The staff are so accommodating and just love walking across the street to the Fairway supermarket with every imaginable goodies to eat (if you choose not to go out for every meal!)", "authors": [“username": "Maureen V", "num_reviews": 2, "num_cities": 2, "id": "E3C85CA9DBBBC77E0DB534ABE93E4713", "location": "Sydney, New South Wales, Australia”], "date_stayed": "December 2012", "offering_id": 93338, "num_helpful_votes": 0, "date": "December 17, 2012", "id": 147639004, "via_mobile": false]
3	[“ratings": [“service": 4.0, "cleanliness": 5.0, "overall": 4.0, "value": 4.0, "location": 5.0, "sleep_quality": 4.0, "rooms": 4.0], "title": "\u201cGreat Stay\u201d", "text": "This is a great property in Midtown. We two different rooms different rooms during our stay. The first room was in the North tower, which was quite inconvenient. You have to go through the conference area to get to the north elevators. \nThe second room was the Andaz Suite. It was nicely appointed room, but the best part about it was the bathroom. From the foot soaking bowl to the bath products, everything about the bathroom was awesome!\nLemon poppy-seed pancakes are must haves at the restaurant. One of the best pancakes ever.", "authors": [“username": "vuguru", "num_cities": 12, "num_helpful_votes": 17, "num_reviews": 14, "num_type_reviews": 14, "id": "FB1032DECE1162CB3556D05F278AAFFD", "location": "Houston”], "date_stayed": "December 2012", "offering_id": 1762573, "num_helpful_votes": 0, "date": "December 18, 2012", "id": 147697954, "via_mobile": false]
4	[“ratings": [“service": 5.0, "cleanliness": 5.0, "overall": 4.0, "value": 5.0, "location": 5.0, "sleep_quality": 5.0, "rooms": 5.0], "title": "\u201cModern Convenience\u201d", "text": "The Andaz is a nice hotel in a central location of Manhattan. The Hyatt has come up with a modern hotel that is both comfortable and convenient. When you arrive you are greeted by friendly \"Hosts\" and they walk you to the check-in desk while offering you a beverage. \nWe had a one bedroom suite that accommodated four people reasonably well. Plenty of closet space, well lit with floor to ceiling windows, and actually quiet!\nThe bathroom was large with a very nice walk-in shower and a built-in bench with unique low spout to wash your feet.\nThe kitchenette was a nice touch with a stocked fridge offering complimentary non-alcoholic beverages and snacks, dishes and utensils, a sink, dishwasher, and a microwave. \nThey have daily Happy Hour(s) where you can get a complimentary decent glass of wine in the modest Lobby Lounge and bring it to your room. The Lobby Lounge has some seating and one table with 8 or so chairs where you can buy food from the adjacent restaurant. One suggestion is to offer selections of cheese and crackers platters to go with that wine. We ordered one that had to be custom made, but it worked well. \nWe didn't eat in the hotel restaurants. The restaurant off the lobby was very pricey for breakfast (+$20 per entree). When you can get a decent

	breakfast within a block or two of the hotel for under $10, what can I say?\nAs a hotel designer.", "authors": [“username": "Hotel-Designer", "num_cities": 5, "num_helpful_votes": 26, "num_reviews": 5, "num_type_reviews": 5, "id": "EC3E275EE7590694889C8C7EE0D13961", "location": "Laguna Beach, CA”], "date_stayed": "August 2012", "offering_id": 1762573, "num_helpful_votes": 0, "date": "December 17, 2012", "id": 147625723, "via_mobile": false]

We start to transform the raw dataset from JSON form into a table by only selecting users’ ratings, users’ reviews and date of comments. Moreover, we change usernames and hotel names to numeric form that replaces both usernames and hotel names by starting with 0 for comfortable interpretation and we select users who have rated and reviewed at least 10 records as shown in Table 6 which shows only the first ten of the records that are transformed to the table. From this, we have got 48,919 records and there are 3,084 users and 3,145 hotels.

Table 6. The raw dataset that transform JSON into table

User	Item	title	Comment	Date	Overall
0	1019	First class service	I am concerned to read several negative comments posted by previo...	April 10, 2010	5
0	1022	Nothing has changed...	Our second stay at this place since 2007 and nothing has really changed...	April 9, 2012	3
0	1022	Nice not Great!!!	My wife and two children stayed here in May as we needed a place to call...	May 22, 2007	3
0	1024	Best Hotel in...	I have been to Seattle many times and have stayed at many fine hotels...	March 24, 2012	5
0	1727	It was an average stay...	I have stayed at this hotel with family on the Regency Club for the past...	July 13, 2012	3
0	1727	A Great Family Hotel...	This hotel is perfect for famalies however for the single business trav...	July 15, 2011	5
0	1727	A great place to stay...	"This is a great hotel perfect for famalies with children. The pool area...	July 27, 2010	5
0	2345	The Club Floor ain't ...	We stayed at this hotel for one night in preparation for our flight the next day...	August 1, 2008	2
0	2625	Fabulous Hotel	Stayed five days while on business in San Francisco. Was upgraded to a cor...	February 24, 2009	5
0	2943	Beware of this conferen...	This hotel caters largely to conference guests and makes no attempt to ser...	September 20, 2005	2

After the dataset has been transformed and presented in the table, we created our 2 input matrices. For the first matrix, we created the matrix that the row represents users, while the column represents hotels and the value in the matrix represents ratings that a user has rated the hotels as shown in Figure 19. If one user rates rating more than one time, we average these ratings. In case users does not rate the hotels yet, rating in that field is filled by 0. We call this matrix as user-hotel rating matrix. For example, Table 7 shows partial user-hotel rating matrix in dataset that is created from Table 6 by using users, items and overall columns. The dimension of user-hotel rating is 3,084 ×3,145 matrix.

$$\begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_n \end{array} \begin{array}{c} \begin{array}{cccc} h_1 & h_2 & \cdots & h_j \end{array} \\ \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1j} \\ r_{21} & r_{22} & \cdots & r_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ r_{\mathrm{i}1} & r_{\mathrm{i}2} & r_{\mathrm{i}3} & r_{ij} \end{bmatrix} \end{array}$$

Figure 19. User-hotel rating matrix

Table 7. The user-hotel rating matrix in dataset

hotel / user	31	32	33	34	35	36	37	38	39	40
18	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0
21	0	4	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0
27	0	0	5	0	0	0	0	0	0	0

Another matrix that represents reviews for each hotel is created. Users' reviews of each user are gathered and grouped by hotel levels. From Figure 20, each hotel has review vector that users in dataset provided to those hotels. We called this matrix as hotel-review matrix. It is used to be input of LDA. AS for Table 8, it shows 3 sample hotels that their reviews from all users in all period were gathered, each hotel's reviews are gathered from every user who provided reviews to this hotel. In this proposed method, there are 2 hotel-review matrices that are used to be input of LDA for extract value from contextual information of users. The first matrix, hotel reviews are gathered from all user reviews given to each hotel. Another matrix, hotel reviews are gathered from user reviews that have reviewed recently. We selected reviews from the user reviews during 2011-2012 in the dataset because these are the latest period of the dataset. The dimension of each hotel-review matrix is 3,145×2 matrices. We will explain the utility of 2 matrices at a later stage.

$$h_1 \quad [Review_{u_1}, Review_{u_3}, Review_{u_7}]$$

$$h_2 \quad [Review_{u_2}, Review_{u_4}]$$

$$h_3 \quad [Review_{u_2}, Review_{u_5}, Review_{u_8}, Review_{u_9}]$$

$$\vdots$$

$$h_n \quad [Review_{u_1}, Review_{u_2}, Review_{u_8}, Review_{u_{10}}]$$

Figure 20. The hotel-review matrix

Table 8. The hotel-review matrix in dataset

Hotel	Reviews
0	Perfect for a couples overnight!! We were 1st time visitors to NYC & opted for this "local flavor" instead of a standard hotel. You feel more like a native than a visitor while staying on the Upper East Side, just 4 1/2 blocks from Central Park. Less than 10 minutes in a cab & you're in the Theatre District.We stayed in 4A & LOVED it. On the 4th floor, includes kitchenette w/ stove, dorm fridge, microwave, toaster, cookware, plates & utensils. Easy to make yourself breakfast or dinner if you want to stay in. Bedroom is large, has AC & the bed is comfy. Bathroom is small but not cramped. Sitting room between the kitchen & bedroom includes a chaise & TV.Clean & quiet room!! The only complaint I had was no internet,

	although I was told that's available on the 1st floor only. Never met the staff, nor did I need their services. Security is not an issue as the property's front door has a passcode & each room has its own key.We would defintely stay here again!!
1	Just back from a 3 night stay at this hotel and had an amazing time in NYC. This hotel is so close to everything, we walked to Times Square, Broadway, the famous 5th Ave was around the corner. Rooms are basic but very clean had no complaints and the staff at reception were very friendly. Would definitely recommend this hotel and will be staying here again if I go back to NYC. Will be recommending to all my friends :), We (an Australian couple aged in our 50's) recently spent 6 nights here. Never having been to NYC before, we greatly enjoyed being in such a central location - it is easy walking distance to Times Square, Broadway theatres, Central Park, etc. Close to a subway station so also easy to get anywhere else in Manhattan. The hotel is the Women's National Republican Club and lots of club activities and other events such as wedding receptions take place here. The Club members we met were all gracious and charming and the staff were friendly and very helpful. We diplomatically kept our political opinions to ourselves. One evening we had the unexpected pleasure of listening to songs being performed very professionally in the dining area by a group of mature-aged people who meet there regularly. Be prepared to wear something dressier than jeans and sneakers in the bar and dining room as there is a dress code, however everyone is so polite I suspect nothing would be said whatever you wore. The building was built in the 1930s and while it is showing signs of age, it is still lovely with lots of wood and marble. Very clean, old- fashioned place. The restaurant meals were delicious. Only disadvantages are that the restaurant is closed on weekends (but there are lots of other places to eat nearby) and the bedrooms, which have single pane windows, get noise coming from the street all night long. On the Saturday night when there was a large event happening, we also experienced some noise coming through clearly from the next room. The noise didn't worry me as I wore ear plugs but it bothered my husband. This is an interesting and comfortable place, more of a club with some guest bedrooms than a normal hotel. Our experience was extremely positive overall and I recommend this Club to anyone who wants something a bit different to a stay in a boring, standard, modern hotel.
2	Very small, very intimate place. The guys who run it are very friendly and accomodating. The rooms are nice and have a boutique feel. You don't have all that extras that would come with a larger hotel, but that's okay. Wish it was a little less expensive. Nice, comfortable. Because it's so small there's a very nice personal and intimate quality to staying here that you wouldn't get anywhere else. It's one of a kind in NYC. Recommended. Only ten minutes from times square, the 414 is in a great location. Rare to have breakfast included in an NYC hotel so we loved it! Walls could do with a tiny lick of paint and the air conditioning was quite noisy but in general the hotel is in great condition and the staff were all lovely. Would definitely recommend.

3.2 Finding missing user s' ratings

In this step, we want to complete user-hotel rating matrix because fields contained a lot of 0 and data in the user-hotel rating matrix is very sparse in our input matrix as examples in Table 7. We apply LDA to solve this problem.

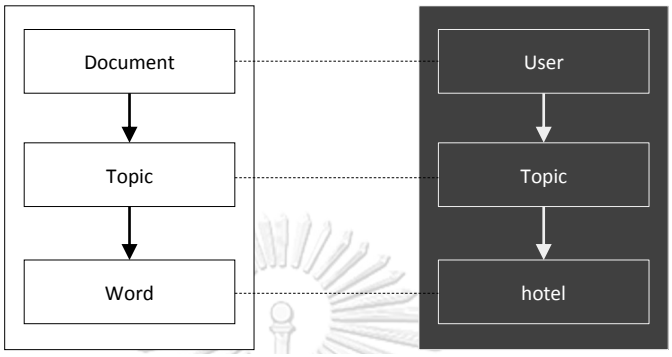


Figure 21. The analogy comparing LDA principle and the user-hotel rating matrix

We start to find missing users' ratings from our input, user-hotel rating matrix in Table 17, by applying LDA. From principles of LDA, users act as documents and hotels act as words as shown in Figure 21. Users' ratings are used as frequency of words contained in the hotel. The number of topics in LDA is set to 20 topics. We get the output from LDA that is user-topic distribution (θ) of which dimension is 3,084×20 matrix and topics-hotels distribution ($\emptyset$) of which dimension is 20×3,145 matrix being shown as examples in Table 9 and Table 10 respectively.

Table 9. The user-topic distribution (θ) from user-hotel rating matrix

Topic / User	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
0	0.001818	0.001818	0.001818	0.001818	0.001818	0.001818	0.001818
1	0.026829	0.002439	0.002439	0.026829	0.002439	0.002439	0.002439
2	0.002174	0.002174	0.002174	0.002174	0.263043	0.002174	0.002174
3	0.002326	0.002326	0.25814	0.118605	0.002326	0.002326	0.281395
4	0.001099	0.089011	0.001099	0.001099	0.023077	0.001099	0.001099

Table 10. The topic-hotel distribution (Ø) from user-hotel rating matrix

Hotel / Topic	0	1	2	3	4	5	6
Topic 0	1.23E-06	1.23E-06	1.23E-06	1.23E-06	1.23E-06	1.23E-06	1.23E-06
Topic 1	9.39E-07	9.39E-07	9.39E-07	9.39E-07	9.39E-07	9.39E-07	9.39E-07
Topic 2	8.96E-07	8.96E-07	8.96E-07	8.96E-07	8.96E-07	8.96E-07	8.96E-07
Topic 3	9.88E-07	9.88E-07	9.88E-07	9.88E-07	9.88E-07	9.88E-07	9.88E-07
Topic 4	1.04E-06	0.000939	1.04E-06	1.04E-06	1.04E-06	1.04E-06	1.04E-06
Topic 5	0.000475	9.48E-07	9.48E-07	0.000759	9.48E-07	9.48E-07	9.48E-07

After passing matrix into LDA, new fully completed users' rating matrix calls *NR matrix* is obtained by using user-topic distribution multiplied by topic-hotel distribution as in Equation 13.

$$NR = \theta \times \emptyset \tag{13}$$

Table 11. The new fully users' rating (*NR*) matrix

Hotel / User	31	32	33	34	35	36	37	38	39	40
18	2.50E-06	0.001943	5.31E-05	2.57E-05	0.000382	6.15E-05	0.000325	4.80E-06	5.60E-06	0.000137
19	1.63E-06	0.002158	7.18E-05	1.02E-05	5.63E-05	9.61E-05	7.45E-06	7.98E-05	2.77E-06	0.000156
20	2.34E-06	4.78E-05	4.90E-05	2.38E-05	1.02E-05	2.02E-05	1.70E-05	4.47E-06	5.20E-06	6.42E-06
21	3.55E-05	0.000727	0.001403	0.001063	0.000391	6.73E-05	0.000306	5.27E-05	0.000218	0.000127
22	1.64E-05	0.001057	0.000334	2.51E-05	5.18E-05	2.14E-05	2.07E-05	6.62E-05	5.52E-06	6.81E-06
23	2.69E-06	0.00313	0.000904	2.80E-05	0.000215	0.001494	0.000187	0.000239	0.000253	7.84E-05
24	1.89E-06	0.001547	0.000307	1.52E-05	6.78E-06	0.000443	0.000224	6.45E-05	3.66E-06	6.38E-05
25	5.95E-05	0.000799	0.00168	0.000456	5.57E-06	0.000117	0.000651	2.76E-06	6.42E-05	3.71E-06
26	0.000124	0.003155	0.002152	0.000768	0.00029	1.38E-05	0.00027	3.38E-06	3.87E-06	0.000104
27	9.15E-05	0.001085	0.003903	2.03E-05	8.88E-06	1.74E-05	3.25E-05	3.99E-06	0.000607	5.65E-06

From this equation, we get new users' ratings to fulfill the missing users' rating in the user-hotel rating matrix. NR is the new matrix which occurs from user-topic distribution (θ) multiplied by topics-hotels distribution ($\emptyset$) as shown as the example of NR matrix in Table 11 and the dimension of *NR* matrix is 3,084 ×3,145 matrix that has as same dimension as user-hotel rating matrix from Table 7. This step applies content-based technique to find missing users' ratings.

3.3 Finding the similarity of users

In this step, we find the similarity of the target users with other users in dataset by using Pearson correlation and forming set of neighbors of the target users to be used in the next step to find potential of collaborative filtering technique in hotels recommendation.

The Pearson correlation coefficient is a measure of the strength of the linear relationship between two variables. It is referred to the correlation coefficient. In our method, it is used to find the similarity between target users and other users. It is used to compare each pair of users until we compare them with all users in the dataset. Each pair of users in our dataset is calculated as in Equation 14. The correlation coefficient of each pair of users has range between -1 to 1. -1 indicates a perfect negative linear relationship between users, 0 indicates no linear relationship between users, and 1 indicates a perfect positive linear relationship between users.

$$Pearson(u,v) = \frac{\sum_{k \in I_u \cap I_v}(r_{uk}-\mu_u)\cdot(r_{vk}-\mu_v)}{\sqrt{\sum_{k \in I_u \cap I_v}(r_{uk}-\mu_u)^2}\cdot\sqrt{\sum_{k \in I_u \cap I_v}(r_{vk}-\mu_v)^2}} \quad (14)$$

u, v are user u and user *v*

I_u is hotel that user *u* used to rate

I_v is hotel that user *v* used to rate

r_{vk} is rating of the hotel that user v rated and user u used to rate this hotel

The *NR* matrix is used to be input for applying Pearson correlation to find similarity on every pair of users. The output matrix has 3084×3084 matrices and the example of output is shown in Table 12. It shows that rows and columns represent users and values are correlation coefficient of each pair of users. Since the value of Pearson correlation is between -1 to 1, we select pairs of users that have correlation coefficient more than 0. Then we find that the number of users who have the least pair that have correlation coefficient more than 0 in dataset is 126 users (neighbors) so the value of correlation coefficient of every user is sorted by being descended and we only select the top 125 users because the users' own pair is discarded. For example, if user number 0 find the similarity with user number 0, the correlation coefficient is always 1 so we discard these kinds of pairs of users. Finally, we get neighbor lists of each user that has similar preferences with them as shown in the example in Table 13. This step applies collaborative filtering technique to find neighbor of target users.

Table 12. The similarity between pair of users from *NR* matrix

User / User	0	1	2	3	4	5	6	7	8	9	10
0	1	0.247549	0.403928	-0.01074	0.039509	0.012913	0.356156	0.014193	0.099674	0.037972	0.786679
1	0.247549	1	0.161987	0.044822	0.090665	0.095205	0.415583	0.397498	0.290271	0.432017	0.231774
2	0.403928	0.161987	1	0.008773	0.060305	0.081852	0.183975	0.288683	0.365594	0.025562	0.361671
3	-0.01074	0.044822	0.008773	1	0.176739	0.094826	0.042487	0.482248	0.151467	0.092538	0.187089
4	0.039509	0.090665	0.060305	0.176739	1	0.037141	0.169699	0.298367	0.341604	0.180597	0.079613
5	0.012913	0.095205	0.081852	0.094826	0.037141	1	0.018617	0.304989	0.1765	0.054659	0.157935
6	0.356156	0.415583	0.183975	0.042487	0.169699	0.018617	1	0.073809	0.354317	0.056838	0.389629
7	0.014193	0.397498	0.288683	0.482248	0.298367	0.304989	0.073809	1	0.364364	0.389768	0.27776
8	0.099674	0.290271	0.365594	0.151467	0.341604	0.1765	0.354317	0.364364	1	0.437915	0.130108
9	0.037972	0.432017	0.025562	0.092538	0.180597	0.054659	0.056838	0.389768	0.437915	1	0.231605
10	0.786679	0.231774	0.361671	0.187089	0.079613	0.157935	0.389629	0.27776	0.130108	0.231605	1
11	0.063327	0.196656	0.217769	0.286301	0.509173	0.04073	0.227397	0.559604	0.696506	0.478647	0.085764

Table 13. The neighbors of each users

User	neighbor top2	neighbor top3	neighbor top4	neighbor top5	neighbor top6	neighbor top7
0	2817	2304	2950	769	2849	1229
1	452	1740	1128	1370	3006	2337
2	1147	2239	1900	1986	423	410
3	3065	2837	2372	1745	2197	2855
4	2457	1601	389	248	2512	1981
5	2374	836	85	819	2934	448
6	2537	860	421	2816	2969	3020
7	1393	3042	2354	313	2870	957
8	305	2912	2333	459	548	2218
9	183	2935	1713	2490	2978	2294
10	2788	725	1774	2771	232	2039
11	737	678	2214	1356	2761	1011
12	729	1217	2558	2416	845	700
13	300	2586	3072	855	943	2490
14	964	231	3062	2911	1169	800
15	1196	645	2750	2959	1818	2460
16	1549	1165	2350	2356	570	2558
17	3002	837	62	226	2027	577
18	1357	1809	973	2664	2916	2764
19	2288	2066	823	2004	462	1136
20	1936	2470	3013	2012	1700	305

3.4 Incorporating context-driven to recommend hotel

In this step, we want to create the recent preference of hotel lists that match with target users' preferences by incorporating context-driven. As for recommender system structure in Figure 18, this part is in dashed lines rectangle. This part, we apply context-driven for improving recommendation. To be more understandable, we provide some examples for this step. The first one, hotels of target users that target users provided high ratings from Table 7 are selected and all of the users' reviews are gathered to get hotel-review matrix as Table 8 and this hotel-review matrix is passed into LDA to get hotel-topic distribution (θ) and summarized to be target user-topic distribution. This is the result that we get target users' preferences profile. It will be used to find hotels that target users may recently like. The second one, we find the hotel preferences which may currently be liked by target users as our assumption that the hotel preferences can be changed all the time, so we choose hotels that have present hotel preferences that match with target users' preferences. As we get neighbors of target users' from step 3.3, for example target user A has user B, user C, user D as neighbor, each neighbor also has predicted rating from NR matrix from step 3.2 so we select top 10 hotels that get highest predicted rating from NR matrix and extract these hotels to be the second hotel-review matrix that gathers users' reviews that already reviewed recently by neighbors. Then it is passed into LDA and the hotel-topic distribution (θ) is obtained. These are used to be hotel preferences that recently like by neighbors. Lastly, after we get the target users' preference profile and recent hotel preferences by neighbors, we apply CBF to find the similarity between them and get the list of hotels that target users may recently like in the present. As for detail, we divided this part into 3 steps; 3.4.1 Creating target users' preference, 3.4.2 Finding recent hotel preference from target users' neighbors and 3.4.3 Creating recent hotel recommendation list for the target users.

3.4.1 CREATING TARGET USERS' PREFERENCE PROFILE

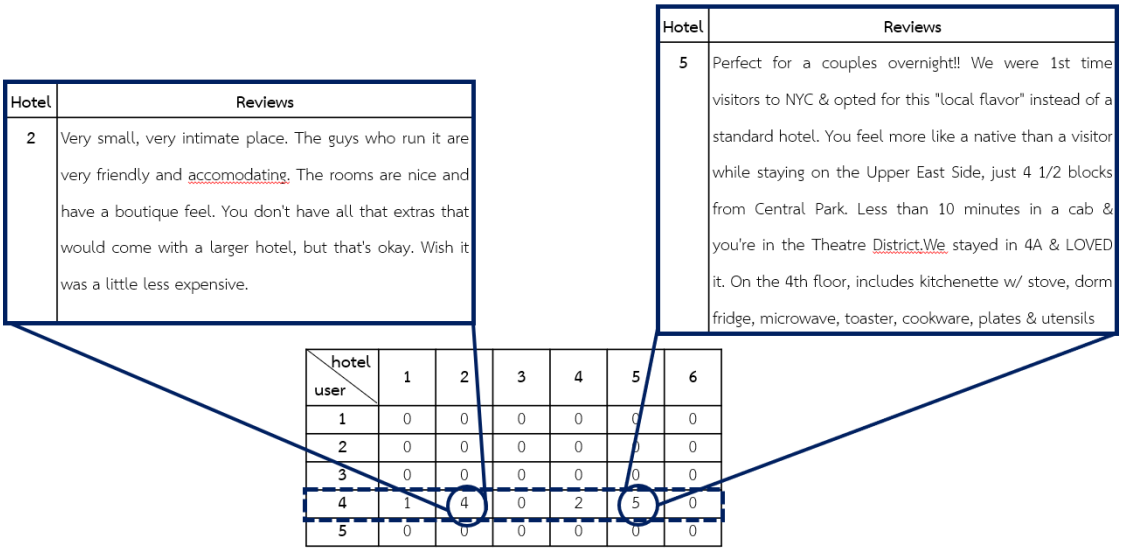


Figure 22. The hotel-review examples extracted from user-hotel rating matrix

From the user-hotel rating matrix in data preparation step as shown in example in Table 7, sets of hotels of each user that have been rated with the score more than 2 from the user-hotel rating matrix are used to represent target users' preference profile. After we got sets of hotels that users rated with high score, we gathered all users' reviews in dataset of these hotels and created hotel review matrix as in example in Table 8. For examples from Figure 22, we assume user 4 is the target user. We find that hotel 2 and 5 get ratings from the target user more than 2 ratings so we crate hotel-review matrix from these hotels. Then, we applied LDA in order to extract latent

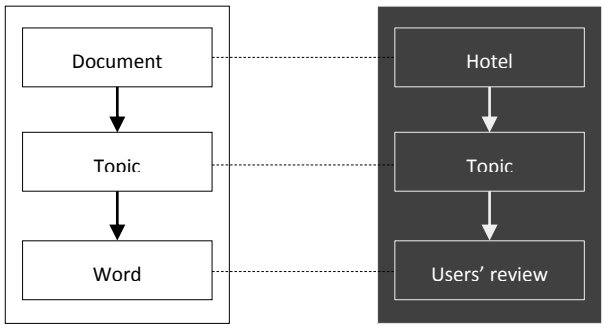


Figure 23. The analogy comparing LDA principle and hotel-review matrix

topic from users' reviews to find hotel-topic distribution (θ) for this set of hotels. The hotels act as documents and the content of users' reviews act as words as in Figure 23.

However, we clean users' reviews before we pass hotel-reviews matrix into LDA model. To be more understandable, we will explain some other steps operating after we pass hotel-reviews matrix from Table 8 into LDA model and explain how we clean users' reviews. Firstly, the users' reviews of each hotel are separated to be singular words by using space. Then, the hotel-word matrix is created. The rows represent hotel, the columns represent singular words and values represent the frequency of word that appear on that hotel. For example, there are 3 hotels, and each hotel contains users' reviews as in Figure 24, and Table 14 shows the examples of transforming hotels' and users' reviews from Figure 24 to the hotel word matrix. The hotel word matrix is the real input that is used for applying the LDA model.

$Hotel_1$ = I love this room. This room is very clean

$Hotel_2$ = this room is prefect

$Hotel_3$= Service is very poor

Figure 24. An example of users' reviews provided to hotels

Table 14. An Example of hotel word matrix

	I	love	this	room	is	very	clean	perfect	Service	poor
$Hotel_1$	1	1	2	2	1	1	1	0	0	0
$Hotel_2$	0	0	1	1	1	0	0	1	0	0
$Hotel_3$	0	0	0	0	1	1	0	0	1	1

From the above example, if we directly pass hotel-reviews matrix in LDA model without transformation and cleansing, its result will not be effective as shown in Table 15. It is an example of topic-word distribution ($\emptyset$). The column represents topic and the row represents the first seventh words and the ratio of each word distributed over

each topic. From the table, we find that some words are meaningless such as und, das, ist etc. in topic 8. We also find some words such as *hotel*. Although it has meaning but it should not be concerned in hotel recommendation because this word cannot refer to any type of hotel explanation or meaningless for users' reviews. For example, the word *clean* and *location*, we can refer these words to the meaning that the hotel is in good location and the room is clean. Moreover, there are some words that have the same meaning, but it appears in other forms. For example, the word *room* in topic 0 and *rooms* in topic 1 have the same meaning, but one is a singular form, and the other is a plural form. From all these problems, it results in affecting some results of topic from LDA that they cannot be interpreted meaningfully.

Table 15. The topic-word distribution from hotel-review matrix before cleansing data

	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7
Topic 0:	hotel	room	service	great	staff	nice	stay
	0.030308	0.024637	0.021788	0.020108	0.015435	0.014082	0.013318
Topic 1:	room	hotel	good	rooms	bed	small	nice
	0.048213	0.024864	0.010745	0.010272	0.010108	0.010091	0.010056
Topic 2:	hotel	staff	great	stay	clean	location	room
	0.035721	0.026333	0.025889	0.022501	0.022403	0.019421	0.018348
Topic 3:	chicago	michigan	location	great	lake	river	view
	0.094602	0.031226	0.028452	0.02204	0.021265	0.020251	0.018312
Topic 4:	airport	hilton	shuttle	hotel	good	flight	service
	0.060978	0.042496	0.037858	0.036657	0.015437	0.014515	0.014138
Topic 5:	room	place	bed	clean	dirty	night	stay
	0.038888	0.023172	0.013235	0.012299	0.010851	0.01059	0.009132
Topic 6:	hotel	historic	rooms	lobby	small	beautiful	building
	0.030323	0.021121	0.019228	0.018983	0.016777	0.015373	0.014148
Topic 7:	hotel	center	nice	great	good	location	marriott
	0.039012	0.017203	0.016773	0.015863	0.015111	0.015111	0.014772
Topic 8:	und	das	die	ist	der	es	zimmer
	0.033245	0.0266	0.026356	0.02207	0.02134	0.014489	0.013347
Topic 9:	hotel	westin	downtown	sheraton	nice	great	dallas
	0.041528	0.040318	0.030182	0.026899	0.021113	0.02019	0.019923

Regarding these problems, we have to improve performance of LDA to increase the effective result as followed. Firstly, we exclude stopping words such as *a, an, the*; Secondly, we change every word to lowercase. Thirdly, we discard all words that are less than 3 alphabets. Fourthly, we change all verb forms to be the infinitive form. Lastly, we exclude some words that should not be contained in reviews such as hotel, stay and good. After we address the problem by applying these solutions, we find that the vocabulary reduces from 62,194 to 6,448. That means the vocabulary is significant and affects LDA interpretation directly as shown in Table 16 which is an example of topic-word distribution from hotel-review matrix after filtering words by the criteria above. From the table, after preprocessing before applying LDA, we find that most of the words in topic-word distribution are significant and help to improve results.

Table 16. The topic-word distribution from hotel-review matrix after cleansing data

	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7
Topic 0:	time	great	clean	subway	central	manhattan	recommend
	0.097183	0.067493	0.04682	0.042685	0.038185	0.024478	0.022515
Topic 1:	time	make	desk	front	great	food	year
	0.031807	0.027099	0.021074	0.016875	0.015316	0.013726	0.01195
Topic 2:	clean	mell	carpet	bathroom	shower	sleep	desk
	0.042146	0.031901	0.029599	0.029048	0.019679	0.018933	0.015302
Topic 3:	breakfast	clean	contin	cereal	great	bagel	avail
	0.21314	0.06956	0.031851	0.028592	0.022424	0.018079	0.014549
Topic 4:	desk	front	tell	clean	time	phone	problem
	0.066301	0.056154	0.044691	0.026433	0.025618	0.024053	0.021954
Topic 5:	clean	breakfast	downtown	drive	freeway	desk	great
	0.064738	0.055161	0.0292	0.028402	0.026994	0.0269	0.025398
Topic 6:	club	sheraton	great	internet	clean	atrium	time
	0.152234	0.133444	0.050076	0.031287	0.028368	0.027912	0.024993
Topic 7:	great	clean	wharf	shop	recommend	window	transport
	0.073393	0.031952	0.031952	0.027946	0.020697	0.020363	0.019362
Topic 8:	chicago	michigan	great	shop	downtown	clean	visit
	0.274346	0.089316	0.086672	0.046066	0.034038	0.027555	0.017147

After applying LDA, we get hotel-topic distribution (θ) vector from sets of these hotels as shown in Table 17. As in Figure 25, we find target users' preference by averaging each topic of this set of hotels. Each topic of hotel vectors is selected and averaged. Finally, each topic of hotel-topic distribution (θ) matrix is merged to be one vector and we use it to be a target user's preference profile as shown in Table 18. For example, user 0 is a target user and user 0 provides hotel high ratings to hotels number 1019, 1024, 1727 and 2625 from Table 17. Thus, these hotels applied LDA and we get hotel-topic distribution of user 0. Then, we average each topic from user 0, for example, at topic 0 in Table 17, the values of topic 0 from user 0 which are 0.00015, 0.000176, 9.39E-05 and 9.41E-05 are averaged to be one value that is 0.000128. Every topic is averaged using the mentioned method until all topics are done. Finally, we get a user 0 preference profile as shown in Table 18.

Table 17. The hotel-topic distribution that hotels get high ratings from users

User	Hotel	Rating	topic0	topic1	topic2	topic3	topic4	topic5
0	1019	5	0.00015	0.490405	0.00015	0.00015	0.00015	0.016642
0	1024	5	0.000176	0.447359	0.003697	0.010739	0.014261	0.021303
0	1727	5	9.39E-05	0.515587	0.104319	0.016995	0.010423	9.39E-05
0	2625	5	9.41E-05	0.50715	0.001976	9.41E-05	0.044309	9.41E-05
1	1043	4	0.01625	0.372132	7.35E-05	7.35E-05	7.35E-05	7.35E-05
1	1047	5	0.049968	0.392356	6.32E-05	0.005749	0.00638	6.32E-05
1	1048	5	0.008171	0.326353	5.41E-05	0.009253	0.023864	0.010335
1	1713	5	0.109035	0.342946	0.078094	0.000124	0.032302	0.001361
1	1715	5	0.026863	0.415086	0.017223	2.61E-05	0.011751	2.61E-05
1	1725	5	0.003614	0.37883	0.000172	0.050086	0.015663	0.029432
1	1728	5	0.065729	0.336563	0.000104	0.000104	0.000104	0.041771
1	2020	4	0.054704	0.195238	0.068641	0.000116	0.000116	0.000116
1	2167	5	0.00012	0.366707	0.001322	0.00012	0.015745	0.00012
1	2744	5	0.000893	0.402679	0.000893	0.000893	0.152679	0.000893
1	2941	5	0.039736	0.385264	0.000102	0.000102	0.019411	0.000102

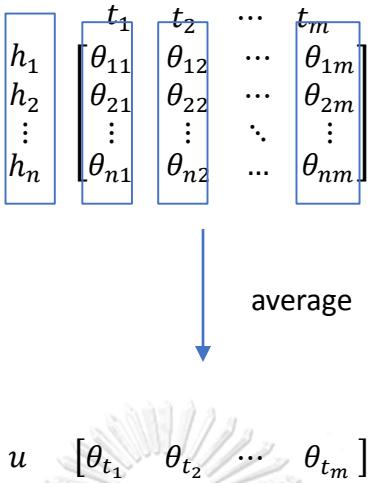


Figure 25. Processes of users' preference profile from hotel-topic distribution

Table 18. The target users' preference profile of each user

User	topic0	topic1	topic2	topic3	topic4	topic5	topic6
0	0.000128	0.490125	0.027535	0.006995	0.017285	0.009533	0.004119
1	0.034098	0.355832	0.015158	0.006059	0.025281	0.007663	0.018051
2	0.043842	0.355536	0.019989	0.000768	0.005824	0.009882	0.017612
3	0.011558	0.290408	0.024495	0.033058	0.032755	0.080003	0.01437
4	0.021512	0.339199	0.01671	0.014665	0.015659	0.001079	0.002463
5	0.120809	0.215896	0.016845	0.101336	0.029003	0.030116	0.005487
6	0.102231	0.364877	0.012677	0.005805	0.047262	0.000667	0.008105
7	0.040541	0.254556	0.031797	0.043733	0.025158	0.028106	0.009474
8	0.15333	0.321064	0.010503	0.029218	0.040168	0.001561	0.021243
9	0.074432	0.335383	0.057755	0.037666	0.03715	0.03019	0.00071
10	0.064974	0.383296	0.007429	0.018573	0.026818	0.014519	0.012827
11	0.002419	0.379925	0.006965	0.016837	0.0173	0.049124	0.018958
12	0.063295	0.363072	0.006874	0.015674	0.021749	0.018064	0.012977
13	0.137879	0.314579	0.016031	0.035044	0.064842	0.00328	0.006046

3.4.2 FINDING RECENT HOTEL PREFERENCE PROFILE CREATED FROM NEIGHBORS

In order to corporate context-driven, we are interested in finding recent preference hotels that are currently liked by neighbors.

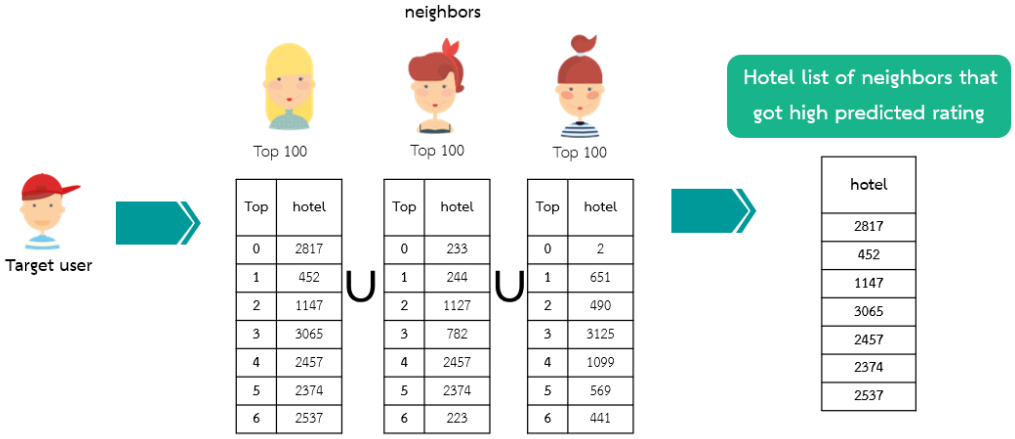


Figure 26. An example of finding hotel list from neighbors

Therefore, we find hotel list of target users by neighbors from step 3.3. From content-based filtering technique in step 3.2, we find the hotel list that the neighbors may like from predicting rating users in *NR* matrix and chose top 100 hotel profiles that are the top 100 of the highest predicted ratings of each neighbor. After that, we get the top 100 hotel profiles from each neighbor of target users and create these hotel lists by including all hotels from the neighbors. Figure 26 shows the process of finding the hotel lists that get high predicted ratings in NR matrix from neighbors.

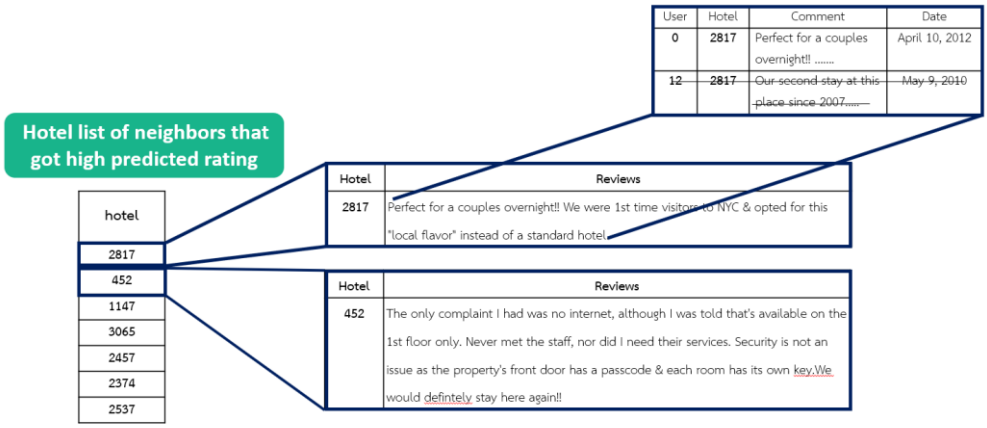


Figure 27. An example of finding hotel-review matrix from neighbors

After that, we create the hotel-review matrix which collects only recent users' reviews by selecting hotel levels from the hotel lists that get high predicted ratings from neighbors and we get hotel-review matrix as in Table 8 from data preparation step. However, reviews in this step that are grouped from users are different from reviews in step 3.4.1. In this step, users' reviews that are grouped in hotel levels were only created in 2012 because these users' reviews are the latest years of dataset because we assume that these users' reviews can represent the present context of hotels. For example, there are 2 users that provide reviews to hotel 2817 from Figure 27. The users' review from user 0 was only collected for creating hotel-reviews matrix because it was created in 2012. Then, we put it into LDA and use principle as shown in Figure 23. We obtain hotel-topic distribution (θ) of target user's neighbor as in Figure 28. Finally, We get the hotel profiles of all hotels which recently liked by target user's neighbors. For example, user 0 is a target user so the neighbors of user 0 are selected from step 3.2 as shown in Table 13. Then, neighbors of user 0 are found including top hotels from *NR* matrix by sorting from high predicted ratings, downwards, and the recent hotel-review matrix and reviews of each hotel are only selected to find hotel-topic distribution of all hotels. Finally, user 0 gets hotel-topic distribution of target user's neighbors as in Table 19.

$$\begin{array}{c} \\ h_1 \\ h_2 \\ \vdots \\ h_n \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \cdots & t_m \end{array} \\ \begin{bmatrix} \theta_{11} & \theta_{12} & \cdots & \theta_{1m} \\ \theta_{21} & \theta_{22} & \cdots & \theta_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \theta_{n1} & \theta_{n2} & \dots & \theta_{nm} \end{bmatrix} \end{array}$$

Figure 28. The hotel-topic distribution (θ) matrix of target user's neighbors

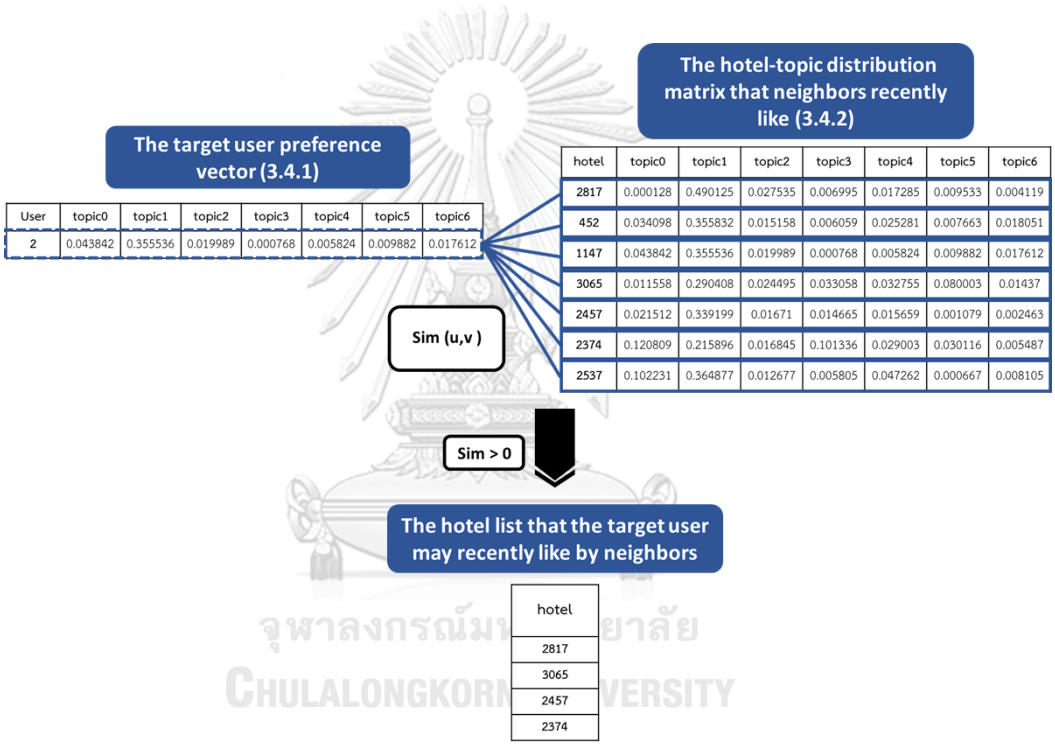
Table 19. The hotel-topic distribution (θ) of target user's neighbors

User	neighbors	Hotel	topic0	topic1	topic2	topic3	topic4	topic5
0	156	2575	0.063269	0.000162	0.16521	0.032524	0.000162	0.000162
0	228	3022	0.000151	0.000151	0.082982	0.000151	0.003163	0.0875
0	1146	2577	0.018664	0.00023	0.046313	0.004839	0.00023	0.00023
0	345	2769	0.069465	0.000122	0.037835	0.000122	0.000122	0.012287
0	689	1022	0.00015	0.022673	0.063213	0.070721	0.015165	0.003153
0	116	2884	0.000186	0.000186	0.022533	0.063501	0.000186	0.026257
0	985	3068	0.002549	0.000121	0.079005	0.000121	0.024393	0.015898
0	443	1553	0.219747	0.000158	0.033333	0.087046	0.000158	0.000158
0	2045	2900	0.000146	0.000146	0.06886	0.008918	0.000146	0.03231
0	3000	2759	0.00025	0.00025	0.12275	0.00025	0.03025	0.07275

3.4.3 FINDING RECENT HOTEL RECOMMENDATION LIST OF THE TARGET USER

After we get target user's preference profile from step 3.4.1 and hotel's preferences profile that recently like by neighbors from step 3.4.2, we apply Pearson correlation to find similarity between target user's preference profile and such hotel's profile. If any hotel's profile has correlation coefficient more than 0, it would be added into hotel list of target user and call the hotel list that target users may recently like by neighbors. Figure 29 shows the example of finding the hotel list that the target user may recently like by neighbors. The target user's preferences of user 2 obtained from step 3.4.1 and hotel's preferences profile that recently like by neighbors of the target user 2 obtained from step 3.4.2. The target user's preferences is compare with each hotel's preferences profile to find the similarity between each other and the hotels that have similarity more than 0 as for hotel number 2817, 3065, 2457 and 2374 are

added in the hotel list that target user 2 may recently like by his neighbors. Table 20 is showing the result between the target user's preference profile and hotel's preferences profile by applying Pearson correlation. The table shows user 0, hotels from neighbor of user 0, and similarity between the target user and hotels. From the table, the similarity that is more than 0 are collected into the hotel list that the target user 0 may recently like by neighbors such as hotel number 1015, 1016, 1022, 1259 and etc. Finally, we obtain the hotel list that target user recently like by neighbors.



User	topic0	topic1	topic2	topic3	topic4	topic5	topic6
2	0.043842	0.355536	0.019989	0.000768	0.005824	0.009882	0.017612

hotel	topic0	topic1	topic2	topic3	topic4	topic5	topic6
2817	0.000128	0.490125	0.027535	0.006995	0.017285	0.009533	0.004119
452	0.034098	0.355832	0.015158	0.006059	0.025281	0.007663	0.018051
1147	0.043842	0.355536	0.019989	0.000768	0.005824	0.009882	0.017612
3065	0.011558	0.290408	0.024495	0.033058	0.032755	0.080003	0.01437
2457	0.021512	0.339199	0.01671	0.014665	0.015659	0.001079	0.002463
2374	0.120809	0.215896	0.016845	0.101336	0.029003	0.030116	0.005487
2537	0.102231	0.364877	0.012677	0.005805	0.047262	0.000667	0.008105

Figure 29. An example of the hotel list that the target user may recently like

Table 20. The similarity result between the target user profile and hotel profile

User	Hotel	similarity
0	1015	0.010367
0	1016	0.007928
0	1017	-0.06164
0	1022	0.028196
0	1023	-0.1568
0	1043	-0.21472
0	1045	0.061065
0	1047	-0.19044
0	1048	-0.09144
0	1168	-0.10991

3.5 Recommending hotel to target users

In this step, we have a set of hotels that target users may recently like from neighbors from step 3.4 that used CF technique and a set of Top 100 hotels that target users have high predicted user ratings of target user from *NR* matrix from step 3.2 that used CBF technique. In this step, we want to find recommended hotel list that target user like in the present by using both sets of hotels, it has been found that there are some parts that are overlapping which means that there are some hotels belonging to both sets. It is selected to recommend the target user. This step is recommended by using content-based filtering technique. Figure 30 shows the example of recommending the recommendation list of the target user. The target user has hotel number 1022, 452, 1147, 3065, 2457, 2374 and 2537 from *NR* matrix that gets the top high predicted ratings. The information was found overlapping as the hotel list that got high predicted rating of target user and the hotel list that the target user 0 may recently like from step 3.4 have got some information that is the same which are hotels number 236,452, 100, 3065, 2415, 202 and 2566. The intersection between 2 hotel lists is applied. Finally, hotel number 452 and 3065 are recommended to target users. That

is the hotel list that target user like in the present. It can be also concluded that the context -driven is added into recommendation list.

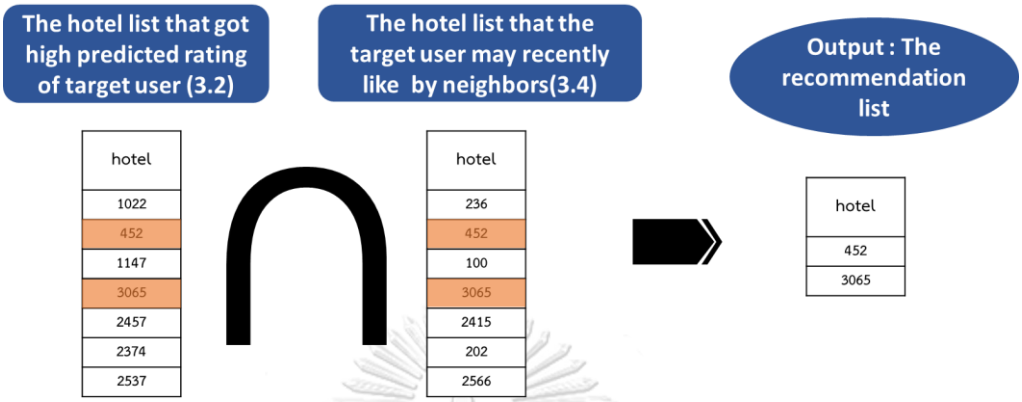


Figure 30. An example of the recommendation list of the target user

CHAPTER 4
EXPERIMENTS AND RESULTS

In this chapter, we describe the experiment that we have conducted in our proposed method and we explain evaluation matrix that we used to evaluate the proposed method comparing with our baselined method. We also show the result that we have got from the experiment and the evaluation matrix. This chapter is divided as follows.

4.1 Data Set

We evaluated our algorithms on the TripAdvisor data set. This raw data set consists of 878,561 users' reviews and ratings (1-5) from 3,084 users on 4,333 hotels. We started by cleaning data set before using them in the evaluation. We select users who have rated and reviewed more than 10 hotels and every hotel that users have rated must also get their reviews. We get 48,920 users' reviews and ratings (1-5) from 3,084 users on 3,145 hotels after we cleaned the data set. There were 48,919 transactions. We used these to evaluate our proposed method and baseline method.

4.2 Baselined method

We compared the result from our proposed method with 2 baseline methods including recommendation by CBF technique integrating with LDA [12] and recommendation by CF technique integrating with LDA [9] on the same data set. Both compared methods are already explained in related work section in Chapter 2.

4.3 Evaluation matrix

We evaluate our proposed method and baselined methods by using normalize Discounted Cumulative Gain (*nDCG*). Normalized Discounted Cumulative Gain (*nDCG*) is a family of ranking measures and is widely used in applications [14]. It is a normalization of the Discounted Cumulative Gain (*DCG*) measure. DCG measure gain of a document is based on its position in the result list. The gain is accumulated from the top of the result list to the bottom, with the gain of each result discounted at lower ranks. *DCG* is a weighed sum of the degree of relevancy of the ranked items as in Equation 15.

$$DCG_p = \sum_{i=1}^{p} \frac{rel_i}{log_2(i+1)} \quad (15)$$

p is a particular rank position

rel_i is the graded relevance of result at position *i*

Our proposed method and baselined methods cannot use *DCG* alone to measure performance, so the cumulative gain at each position for a chosen value of *p* should be normalized across queries. This is done by sorting all relevant documents in the corpus by their relative relevance, producing the maximum possible DCG through position *p*, also called *Ideal DCG* (*IDCG*) through that position. Finally, we can compare our proposed method and baselined method performance by using *nDCG* as in Equation 16.

$$_nDCG_p = \frac{DCG_p}{IDCG_p} \quad (16)$$

The result of *nDCG* can be interpreted following range 0 to 1. The result that is close to 1 means that the algorithm is very efficiency because that result is quite as same as the target user's preference. On the other hand, the result that is close to 0 means that the result is not as same as the target user's preference. The algorithm has poor performance.

From $nDCG_p$ measurement, firstly, we find the recommendation list obtained from our proposed method. We sort the hotel recommendation list by using predicted rating from *NR* matrix. After that, hotels that are recommended from our proposed method are multiplied by 1000 and they bubble up to the top of the list of target users. We get recommendation list for $nDCG_p$ measurement. Another value to find $nDCG_p$ measurement is $IDCG_p$. This hotel recommendation list is obtained by sorting hotels from all of ratings that target users have been rated in the past. Then, we only select a set of hotels that a target user has been rated in the past in recommendation

list from our proposed method as shown in Table 21, and we use them to calculate $nDCG_p$ from Equation 15. For DCG_p, we rank the hotel recommendation list from RecommendRating of our proposed method and for $IDCG_p$, we rank it from TrueRating of target user. Both ranks are shown in the column Recommend_rank and True_rank respectively. Finally, we get the
$nDCG_p$ of our proposed method.

Table 21. The rank created from real rating predicted rating of target user

User	Hotel	TrueRating	RecommendRating	Recommend_rank	True_rank
0	1022	3	16.155958	1	5
0	2625	5	0.006702794	2	1
0	1019	5	0.005246614	3	1
0	1727	4.3333	0.00418998	4	4
0	1024	5	0.003629854	5	1
0	2943	2	0.002944727	6	6
0	2345	2	0.001117787	7	6
1	1715	5	0.008469623	1	1
1	1043	4	0.006569467	2	10
1	1048	5	0.004037626	3	1
1	2941	5	0.003742621	4	1
1	1047	5	0.003476098	5	1
1	2020	4	0.003221428	6	10
1	2167	5	0.003200439	7	1
1	1713	5	0.002900585	8	1
1	1725	5	0.002618267	9	1
1	1728	5	0.001872238	10	1
1	2744	5	0.000600893	11	1

4.4 Result

In this section, we report the recommendation list from our proposed method and compare this with the recommendation list from both baseline methods by using *nDCG* to measure the performance of all methods on the same dataset.

Table 22. The *nDCG* result for recommendation list in each method

Method	*nDCG*
The proposed recommendation method	0.481520361971823
The baseline by CBF + LDA method	0.481348841957661
The baseline by CF + LDA method	0.359036952342095

The result of *nDCG* showed that our recommendation list from our proposed method got the highest result compared with the recommendation list from both baseline methods as shown in Table 22.

CHAPTER 5
DISCUSSION AND SUMMARY

Our proposed method outperforms other methods because we use integrating CBF and CF with LDA both on users' ratings and users' reviews for recommendation. Moreover, we also apply context-driven to help improve our result by considering about time because time has an impact on user's decision making. Preferences of users in the past may not match with the current preference of users.

Comparing CF integrating LDA with our proposed method as shown in Figure 31, our proposed method gets higher *nDCG* result than CF baseline because they only use LDA on user ratings to find the similarity between target users and neighbors and use similarity between them to create predicted users' ratings. In contrast, our proposed method does not only use LDA on users' ratings to find the similarity between target users and neighbors, but we also use LDA on users' reviews and integrating context-driven to find hotels which target users may like in the present period by presenting current hotels liked by neighbors who have similar preferences to target users.

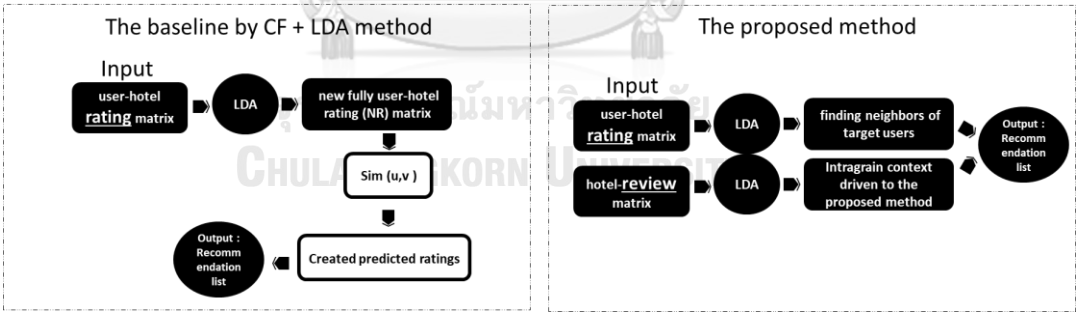


Figure 31. The comparison of CF integrating LDA with our proposed method

Comparing CBF integrating LDA with our proposed method as shown in Figure 32, we get slightly higher *nDCG* result than CBF baseline because both methods find predicted user ratings using LDA on user ratings, where a user acts as a document and a hotel acts as a word. After getting the result from LDA, we multiply users-topics distribution (θ) and topics-hotels distribution ($\emptyset$) to get predicted user's rating result.

Then, CBF baseline method selects top N-hotels which are similar to target user's profile to be the recommendation results. In contrast, we do not only use CBF, we also integrate CF and context-driven for improving recommendation. Thus, we combine current hotels liked by neighbors who have similar preferences to target users from CF part with hotels which are liked by target users from CBF part, where overlapping hotels are bubble up into top of the recommendation list and affect the improved score of our proposed method.

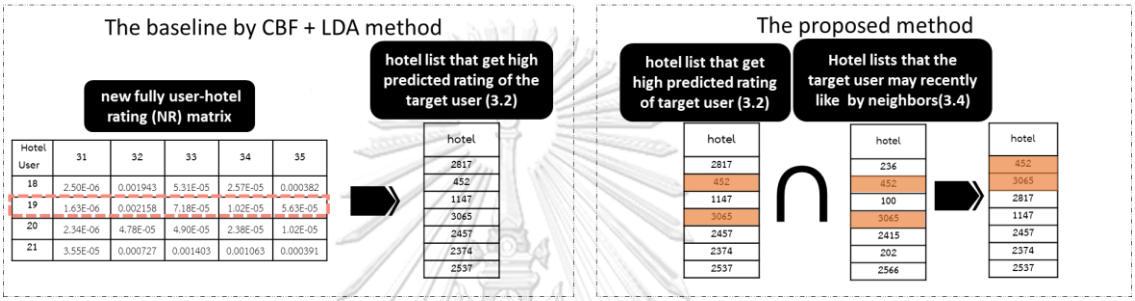


Figure 32. The comparison of CBF integrating LDA with our proposed method

However, the *nDCG* result of CBF baseline and proposed method are very close because our context-driven concept is limited by this dataset. As users' reviews is not enough to present hotels currently liked by neighbors, there are only small numbers of overlapping hotels between neighbors and target users. Therefore, it is not significant in recommendation result.

This paper proposes a hotel hybrid recommendation method based on context-driven using LDA. In our proposed method, we use both LDA on users' ratings and LDA on users' reviews (context). Moreover, we also provide both CF and CBF technique to improve our recommendation. We compare our proposed method to baseline methods that recommend by using either CF or CBF integrating with LDA. We evaluate our proposed method with both baselined methods by using *nDCG*. The result shows that the proposed method achieves outstanding in result accuracy from *nDCG* measurement.

REFERENCES

1. Bazire, M., et al., *Understanding context before using it*, in *Proceedings of the 5th international conference on Modeling and Using Context*. 2005, Springer-Verlag: Paris, France. p. 29-40.
2. Palmisano, C., A. Tuzhilin, and M. Gorgoglione, *Using Context to Improve Predictive Modeling of Customers in Personalization Applications.* IEEE Transactions on Knowledge and Data Engineering, 2008. **20**(11): p. 1535-1549.
3. Adomavicius, G., et al., *Incorporating contextual information in recommender systems using a multidimensional approach.* ACM Trans. Inf. Syst., 2005. **23**(1): p. 103-145.
4. Bulander, R., et al. *Comparison of Different Approaches for Mobile Advertising*. in *Second IEEE International Workshop on Mobile Commerce and Services*. 2005.
5. Adomavicius, G. and A. Tuzhilin, *Context-Aware Recommender Systems*, in *Recommender Systems Handbook*, F. Ricci, et al., Editors. 2011, Springer US: Boston, MA. p. 217-253.
6. Pagano, R., et al., *The Contextual Turn: from Context-Aware to Context-Driven Recommender Systems*. 2016. 249-252.
7. Neammanee, T. and S. Maneeroj. *Time-Aware Recommendation Based on User Preference Driven*. in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. 2018.
8. Blei, D.M., A.Y. Ng, and M.I. Jordan, *Latent dirichlet allocation.* J. Mach. Learn. Res., 2003. **3**: p. 993-1022.
9. Na, L., et al. *Improved Collaborative Filtering Algorithm Using Topic Model*. in *2016 17th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*. 2016.
10. Takuma, K., et al. *A Hotel Recommendation System Based on Reviews: What Do You Attach Importance To?* in *2016 Fourth International Symposium on Computing and Networking (CANDAR)*. 2016.

11. Zhang, J., et al. *An implicit feedback integrated LDA-based topic model for IPTV program recommendation*. in *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*. 2016.
12. Krestel, R., P. Fankhauser, and W. Nejdl, *Latent dirichlet allocation for tag recommendation*, in *Proceedings of the third ACM conference on Recommender systems*. 2009, ACM: New York, New York, USA. p. 61-68.
13. Nagori, R. and G. Aghila. *LDA based integrated document recommendation model for e-learning systems*. in *2011 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC)*. 2011.
14. Jarvelin, K. and J. Kekalainen, *IR evaluation methods for retrieving highly relevant documents*, in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. 2000, ACM: Athens, Greece. p. 41-48.

VITA

NAME	Weraphat Nimchaiyanan (Pete)
DATE OF BIRTH	23 January 1993
PLACE OF BIRTH	Nakhonsawan
INSTITUTIONS ATTENDED	He received a Bachelor's degree in Biotechnology from Mahidol University. Now he is a Master's degree student in Computer Science and Information Technology, Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University.
HOME ADDRESS	174/222 The Tree Interchange Condo, Bangsue, Bamgesue, Bangkok, 10800