THE DESIGN OF A MOBILE ENGINE FOR PERSONALIZED TOURIST ATTRACTION RECOMMENDATION USING SOCIAL NETWORKING SERVICES

Komkid Chatcharaporn

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Information Science in Information Technology

Suranaree University of Technology

Academic Year 2013

การออกแบบเครื่องประมวลผลแบบเคลื่อนที่สำหรับการแนะนำข้อมูลสถานที่ท่องเที่ยวส่วนบุคคลโดยใช้บริการเครือข่ายทางสังคม

นายคมคิด ชัชราภรณ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาการสารสนเทศดุษฎีบัณฑิต
สาขาวิชาเทคโนโลยีสารสนเทศ
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2556

THE DESIGN OF A MOBILE ENGINE FOR PERSONALIZED TOURIST ATTRACTION RECOMMENDATION USING SOCIAL NETWORKING SERVICES

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for the Degree of Doctor of Information Science in Information Technology.

Thesis Examining Committee

(Assoc. Prof. Dr. Weerapong Polnigongit)
Chairperson

(Asst. Prof. Dr. Thara Angskun)
Member (Thesis Advisor)

(Assoc. Prof. Dr. Punpiti Piamsa-nga)
Member

(Assoc. Prof. Dr. Nittaya Kerdprasop)
Member

(Dr. Suphakit Niwattanakul)
Member

(Asst. Prof. Dr. Jitimon Angskun)
Member

(Prof. Dr. Sukit Limpijumnong)
Vice Rector for Academic Affairs and Innovation

(Dr. Peerasak Siriyothin)
Dean of Institute of Social Technology

คมคิด ชัชราภรณ์ : การออกแบบเครื่องประมวลผลแบบเคลื่อนที่สำหรับการแนะนำข้อมูลสถานที่ท่องเที่ยวส่วนบุคคลโดยใช้บริการเครือข่ายทางสังคม (THE DESIGN OF A MOBILE ENGINE FOR PERSONALIZED TOURIST ATTRACTION RECOMMENDATION USING SOCIAL NETWORKING SERVICES) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.ธรา อั่งสกุล, 219 หน้า.

ปัญหาสารสนเทศโหลดเกินเป็นปัญหาที่เกิดขึ้นกับผู้ใช้ในยุคอินเทอร์เน็ตเมื่อไม่นานนี้ และส่งผลกระทบต่อการท่องเที่ยวอิเล็กทรอนิกส์และการท่องเที่ยวแบบเคลื่อนที่ ปัญหานี้ทำให้เกิดข้อมูลที่ไม่มีประโยชน์จำนวนมากบนอินเทอร์เน็ต และเป็นการเพิ่มภาระแก่ผู้ใช้ในการคัดกรองข้อมูลที่ต้องการ อีกทั้งทำให้ผู้ใช้รู้สึกไม่พอใจในที่สุด ดังนั้นระบบแนะนำข้อมูลถูกนำเสนอขึ้นเพื่อแก้ไขปัญหาสารสนเทศโหลดเกินนั้น โดยความสามารถของระบบคือ การวิเคราะห์ความชอบของผู้ใช้จากพฤติกรรมการใช้ระบบในอดีต ความชอบของผู้ใช้นี้สามารถนำไปใช้ในการแนะนำข้อมูลใหม่ที่ผู้ใช้อาจสนใจได้

ในงานวิจัยนี้ได้ออกแบบเครื่องประมวลผลแบบเคลื่อนที่สำหรับการแนะนำข้อมูลสถานที่ท่องเที่ยวส่วนบุคคลโดยใช้บริการเครือข่ายทางสังคม โดยเครื่องประมวลผลนั้นมีการนำวิธีการวิเคราะห์ความหมายแฝงมาใช้ร่วมกับกระบวนการเรียนรู้ของเครื่อง ซึ่งมี 4 เทคนิค ได้แก่ นาอีฟเบย์ส ต้นไม้การตัดสินใจ โครงข่ายประสาทเทียมชนิดแพร่กลับ และซัพพอร์ทเวกเตอร์แมชชีน เพื่อใช้ในการสร้างแบบจำลองการทำนายประเภทของสถานที่ที่ไม่ได้กำหนดไว้ ส่วนกรณีการสร้างแบบจำลองการแนะนำข้อมูลสถานที่ท่องเที่ยวส่วนบุคคล เครื่องประมวลผลใช้เทคนิคการกรองข้อมูลแบบพึ่งพาร่วมที่เน้นผู้ใช้เป็นหลัก ด้วยวิธีแจ็คการ์ดและวิธีโคไซน์

การประเมินเครื่องประมวลผลแบ่งออกเป็น 3 ส่วน โดยส่วนแรก คือ การประเมินความถูกต้องของแบบจำลองในการทำนายประเภทของสถานที่ท่องเที่ยว ส่วนที่สองเป็นการประเมินความถูกต้องของการแนะนำข้อมูลสถานที่ท่องเที่ยว และส่วนสุดท้าย คือ การประเมินเวลาที่ใช้ในการแนะนำข้อมูลสถานที่ท่องเที่ยว โดยการประเมินส่วนแรกใช้ชุดข้อมูลสถานที่ท่องเที่ยวจำนวน 10,250 สถานที่ กับประเภทสถานที่ท่องเที่ยว 11 ประเภทซึ่งได้จากออนโทโลยีชื่อควอลมี จากผลการประเมินพบว่าการใช้เทคนิคโครงข่ายประสาทเทียมชนิดแพร่กลับ และซัพพอร์ทเวกเตอร์แมชชีน โดยใช้ขนาดข้อมูลความหมายแฝงร่วมที่จำนวน 1,200 มิติ ให้ผลลัพธ์การทำนายที่มีประสิทธิภาพดีที่สุด คือ ให้ค่าความระลึกในการทำนายที่ร้อยละ 75.96 และ 77.82 ตามลำดับ

การประเมินส่วนที่สองใช้ข้อมูลประวัติการเช็คอินสถานที่ท่องเที่ยวและข้อมูลเพื่อนในบริการเครือข่ายทางสังคมของผู้ใช้ที่เป็นผู้เข้าร่วมทดสอบจำนวน 15 คน โดยข้อมูลที่ใช้ในการ

สร้างแบบจำลองการแนะนำแบ่งเป็น 4 ประเภท ได้แก่ 1) การเลือกข้อมูลผู้ใช้ทั้งหมดที่มีในชุดทดสอบ 2) การเลือกข้อมูลผู้ใช้เฉพาะที่เป็นเพื่อนในบริการเครือข่ายทางสังคมกับผู้เข้าร่วมทดสอบ 3) การเลือกข้อมูลผู้ใช้เฉพาะที่เป็นเพื่อนซึ่งได้มาจากการคัดกรองข้อมูลทางประชากรศาสตร์ที่มีความคล้ายคลึงกันทั้งหมด และ 4) การเลือกข้อมูลผู้ใช้เฉพาะที่เป็นเพื่อนซึ่งได้มาจากการคัดกรองข้อมูลทางประชากรศาสตร์ที่มีความคล้ายคลึงกันกับผู้ใช้มากที่สุดเพียง 200 คน ซึ่งผลการประเมินแสดงให้เห็นว่าเทคนิคการเลือกเพื่อนบ้านด้วยวิธีแจ็คการ์ดให้ผลลัพธ์การแนะนำข้อมูลโดยรวมที่ดีกว่าวิธีโคไซน์ การใช้วิธีแจ็คการ์ดร่วมกับการเลือกข้อมูลทั้ง 4 วิธี ส่งผลให้เครื่องประมวลผลให้ผลลัพธ์ความถูกต้องในการแนะนำแบบส่วนบุคคลที่ 64.49% 63.68% 64.03% และ 40.53% ตามลำดับ ส่วนวิธีการแนะนำแบบไม่เป็นส่วนบุคคลนั้น เครื่องประมวลผลให้ความถูกต้องอยู่ที่ 35.02% 36.55% 29.7% และ 31.1% ตามลำดับ

การประเมินส่วนสุดท้ายใช้ข้อมูลชุดเดียวกับการประเมินส่วนที่สอง ผลการประเมินแสดงให้เห็นถึงความเป็นไปได้ในการนำเทคนิคการคัดกรองข้อมูลทางประชากรศาสตร์มาช่วยลดระยะเวลาการประมวลผลได้ แต่อย่างไรก็ตามวิธีการเลือกข้อมูลโดยใช้เฉพาะข้อมูลเพื่อนในบริการเครือข่ายทางสังคมให้ผลลัพธ์การแนะนำข้อมูลได้เร็วที่สุด ด้วยวิธีการนี้เครื่องประมวลผลสามารถแนะนำข้อมูลสถานที่ท่องเที่ยวส่วนบุคคลได้ภายในระยะเวลา 2 วินาที

สาขาวิชาเทคโนโลยีสารสนเทศ ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2556 ลายมือชื่ออาจารย์ที่ปรึกษา_____

ลายมือชื่ออาจารย์ที่ปรึกษาร่วม_____

KOMKID CHATCHARAPORN : THE DESIGN OF A MOBILE ENGINE FOR PERSONALIZED TOURIST ATTRACTION RECOMMENDATION USING SOCIAL NETWORKING SERVICES. THESIS ADVISOR : ASST. PROF. THARA ANGSKUN, Ph.D., 219 PP.

PERSONALIZED RECOMMENDATION/SOCIAL NETWORKING SERVICES/ MOBILE ENGINE

Recently, a problem that has been occurred with the users in the Internet era and has also been affected the *e*-Tourism and *m*-Tourism is the information overload problem. This problem delivers lots of useless data from the Internet, and it makes users burden to filter them and causes them to be nervous. Hence, recommendation systems are proposed to overcome the problem. The systems have ability to analyze users' preferences based on their behaviors in the past. The users' preferences are able to be used to suggest new items that the users might interest.

This research designs a mobile engine for personalized tourist attraction recommendation using social networking services. The recommendation engine applies a latent semantic analysis and four machine learning algorithms for constructing prediction models of incomplete attraction categories. Those machine learning algorithms comprise Naïve Bayes (NB), Decision Tree (J48), Back-Propagation Neural Networks (BPNN), and Support Vector Machine (SVM). In case of constructing personalized models for attraction recommendation, the mobile engine applies a user-based collaborative filtering technique to achieve the recommendation process.

The evaluation of the mobile engine is divided into three parts. The first one is the assessment of performance of category prediction. The second one is the evaluation of correctness of recommendation. The other one is the appraisement of response time. The first evaluation is conducted with 10,250 attractions and 11 categories based on QALL-ME ontology. The evaluation results indicate that both SVM and BPNN algorithms with 1,200 dimensions of latent semantic space are the two most efficiency approaches. They are able to provide the performance with 77.82% and 75.96% of recall, respectively.

The second evaluation is performed with datasets of fifteen active users including their check-in histories and SNS friends. Data selection for constructing the recommendation models consists of 4 groups as follows: 1) Selecting all users in the entire dataset; 2) Selecting solely SNS friends of each active user; 3) adopting all demographic filtering (DF)-based friends; 4) applying top-200 DF-based friends. The evaluation results reveal that Jaccard similarity offers better correctness than Cosine similarity. With Jaccard similarity, the correctness of personalized approach based on four data selection is 64.49%, 63.68%, 64.03% and 40.53%, whereas non-personalized approach is 35.02%, 36.55%, 29.7% and 31.1%.

The third evaluation utilizes the same datasets as the second one. The last assessment indicates that DF-approach shows the possible way to decrease the response time of the mobile engine. Nonetheless, using solely SNS friends takes the shortest time to complete recommendation. It can be completed within two seconds.

School of Information Technology Student's Signature _____

Academic Year 2013 Advisor's Signature _____

Co-advisor's Signature _____

ACKNOWLEDGEMENTS

This dissertation could not have been completed without the guidance and support of the kind people around me.

First, I would like to express my sincere gratitude to my research supervisors, Asst. Prof. Dr. Thara Angskun and Asst. Prof. Dr. Jitimon Angskun, for their excellent guidance, enthusiastic encouragement and immense knowledge. Their guidance helps me in all the time of research and writing of this dissertation. I am greatly appreciated for their advice and teaching. Without them, I would not have been able to complete my dissertation.

I would like to express my very great appreciation to Assoc. Prof. Dr. Weerapong Polnigongit, the chairman of my dissertation committee, for his suggestion and assistance. My grateful thanks are also extended to the committees for my dissertation who are Assoc. Prof. Dr. Punpiti Piamsa-nga for his insightful comments, Assoc. Prof. Dr. Nittaya Kerdprasop for helping me keep things in perspective, and Dr. Suphakit Niwattanakul for giving me support and encouragement.

I would also like to extend my thanks to Phatchara Yubon for her help in offering me the resources in running and testing the system.

My grateful thanks go to Institute of Social Technology, School of Information Technology, Suranaree University of Technology for offering me a scholarship in order to study a doctoral degree and to all of my lecturers at the School of Information Technology for their kindness in teaching me throughout my study

Finally, I most gratefully acknowledge my parents, two younger sisters, and my friends. They are always supporting me and cheering me up with their best wishes.

Komkid Chatcharaporn

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF TABLES

Table **Page**

LIST OF TABLES (Continued)

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Introduction

Recently, a problem that has been occurred with the users in the Internet era and has also been affected the e-Tourism and m-Tourism is the information overload problem. The problem had begun when the e-Commerce grew rapidly. In early e-commerce era, many businesses had their own websites in order to provide information to their customers. Thus, there are a lot of business information on the Internet. In addition, the emergence of Web 2.0 services such as blogs, wikis, social networking sites, as well as content sharing sites, which encourage users to create their contents, express their opinions and share their contents to others. Nowadays, these factors cause the Internet to become enormous sources of information. Plenty of information makes users annoy when they are searching data what they are looking for on the web. Moreover, returning lots of useless data from the Internet, it makes users burden to filter them and causes them to be nervous (Bawden and Robinson, 2009, pp. 3-6).

The recommendation systems are mostly used to overcome the information overload problem for tourists in Web 2.0 era (Kabassi, 2010) because the systems can provide information matching individual interests of each tourist. Moreover, the systems are an appropriate solution for the modern tourism due to customization and personalization (Biederman, 2007, pp. 567-568). The personalization implemented

with *m*-Tourism leads many possible ways to offer individual information to travelers on their mobile devices based on contextualizing, localizing and personalizing. Besides, it can also help the travelers to plan and receive actual experiences of travelling. Therefore, the personalization applications and services play a key role in the breakthrough of *m*-Tourism in the future. Because of the changing behaviors of modern travelers, they are expecting the mobile technology to provide them with up-to-dated, location-based and personalized information for improving their travel's experiences (Egger and Buhalis, 2008, p.417).

Most existing recommendation systems were based on user's profiles and his/her preference in the past (Xu, Zhang and Zhou, 2005; Kanellopoulos, 2008; Wang et al., 2008; Lee, Chang and Wang, 2009). Although the recommendation system is developed continually, it still does not provide satisfied recommendation to the users from the beginning because of their limitation. There are many approaches of recommendation system such as demographic filtering, content-based filtering, collaborative filtering, knowledge-based filtering and hybrid approach (Montaner, Lopez and De La Rosa, 2003). However, modern recommendation systems mostly exploit the hybrid approach. Demographic filtering, content-based filtering and collaborative filtering are the major approaches widely adopted in the hybrid approach (Montaner et al., 2003; Fijałkowski and Zatoka, 2011).

Content-based filtering (CBF) has capability to recommend items based on similarity between descriptions of items and user's favorite items in the past (Meteren and Someren, 2000). Therefore, this approach is able to track each user's interests in the system individually. Nevertheless, the approach has some drawbacks. CBF cannot select other types of items which have not been rated from the user. For

instance, if the user never rates a restaurant, it cannot recommend any restaurants to the user. Besides, such overspecialization problem makes the recommendation system feed outcomes which user already knows (Montaner et al., 2003).

Collaborative filtering (CF) is one of common approaches used in the recommendation system. It can solve the limitation of CBF as mentioned above by using other users' opinion for recommendation. The CF performs recommendation based on either user or item. User-based CF suggests unseen items already rated by similar users called neighbors. Item-based CF advises items which have the highest correlation based on previous rated items of the active user (Schafer, Frankowski, Herlocker and Sen, 2007, pp. 301-305). However, there are some shortcomings of this approach. The shortcomings include sparsity, cold-start and scalability. The first one is sparsity. The problem occurs when numbers of users are smaller than numbers of items. The occurrence makes the coverage of ratings becoming too sparse. The sparsity causes the system work harder to find the similar users or items and provides low accuracy results. The cold-start problem occurs when there are new added users or items in the system. The new added items cannot be suggested to the user without any rating. In case of the new added users, they can suffer from low accuracy outputs because they have not rated any items thus the system cannot find the similar users. Another problem is scalability. It causes the recommendation system to conduct expensive computation when it is dealing with large amounts of data (Choi, Cho, Choi, Hwang, Park and Kim, 2009; Wang, De Vries and Reinders, 2008, p.21).

Stereotype or demographic filtering (DF) chooses items based on mutual demographic characteristics of users such as age, gender, education, geographical location, etc. The advantages of DF are that it does not rely on item description and

low system interaction does not affect its recommendation. Hence, the cold-start problem cannot harm the DF approach (Drachsler, Hummel and Koper, 2007, p.23). Unfortunately, this approach also has some drawbacks. Because the DF approach is relied on user profiles, insufficient information may lead to unpleasant results (Koychev, 2000, p.101; Mobasher, 2007, p.127; Rao and Talwar, 2008).

In addition to recommendation approaches, the data acquisition is also associated with quality of recommendation. Typically, acquiring data of recommendation system is divided into explicit and implicit methods. An example of the explicit method is questionnaire inquiry used to gather requirements from users directly. The method does not widely apply with the system because the users may have to answer too many questions. Furthermore, sometimes users cannot describe themselves and their preferences correctly. The implicit method is more reliable and non-intrusive than the explicit method because it implies the user preferences from system interactions of users (e.g. website or system). Nevertheless, the generated hypotheses for each user may not be accurate because the system does not have sufficient time to observe each user. Besides, there is combination of explicit and implicit methods. For example, some recommendation systems adopt the explicit way at the beginning and the implicit way is executed when they have sufficient data. According to Kabassi (2010), a large number of personal data from social networking services (SNSs) are very interesting resources to implement with the tourism recommendation system.

The limitations of solely recommendation approaches as mentioned above decrease the accuracy of recommendation system. Both CBF and CF suffer from the cold-start problem mutually. Although DF can feed recommendation without the

cold-start problem, the approach needs the sufficient user profiles to do analysis. Thus, these solely information filtering techniques cause the recommendation system to provide unpleasant feedbacks to the user from the very beginning. There is a hybrid approach that overcomes drawbacks of solely recommendation techniques by improving performance of recommendation. The SNSs and location-based SNSs (LBSNSs) have become interesting to the researchers in the domain of tourism recommendation system. These web 2.0 services currently are very interesting resources for inferring user preferences, particularly LBSNSs implemented in recent systems in order to provide location recommendations (Kim and Ahn, 2012; Esslimani, Brun and Boyer, 2009; He and Chu, 2010; Ma, Zhou, Liu, Lyu and King, 2011).

There were many work attempting to implement SNSs and LBSNSs with attraction recommendation for travelers (García-Crespo, Chamizo, Rivera, Mencke, Colomo-Palacios, and Gómez-Berbís, 2009; Ye, Yin and Lee, 2010; Lian and Xie, 2011; Xiao, Zheng, Luo and Xie, 2010; Berjani and Strufe, 2011). SPETA introduced by García-Crespo et al. (2009) took SNS named OpenSocial to integrate with a system. In SPETA, user profiles and a contact list of the active user were retrieved to determine his/her preferences. However, there was no mention of recommendation performance and processing time in SPETA. Most of the work chose the famous LBSNS such Foursquare for their research (Ye Yin and Lee, 2010; Lian and Xie, 2011). Some picked the local LBSNS such Dianping as main data set (Lian and Xie, 2011). Berjani and Strufe (2011) selected Gowalla to do a spot recommender. Gathering information on those resources was executed through application programming interfaces (APIs) provided by those LBSNSs. To find the similar user,

they need large amounts of data for covering the social relationship. “Friends” in some attempts were assumed as the similar users in order to enhance the CF approach called Friend-Collaborative-Filtering (FCF) (Ye, Yin and Lee, 2010). A data mining technique, particularly clustering, was widely used in those attempts (Xiao et al., 2010; Lian and Xie, 2011). The purpose of clustering was to group the “Check-in” behaviors and identify mutual interests in each cluster before recommending the attractions. However, the large number of gathered data seems to take a long time and use expensive computation. This made some attempts need to specify regions and require offline-mode to compute the whole data (Berjani and Strufe, 2011; Lian and Xie, 2011).

Besides, there were many endeavors trying to enhance quality of POI recommendation. Most of recent work concentrates on POI recommendation in LBSNSs because check-in histories of users could be represented to tastes of users in a tourism domain. Thus, all of them raised LBSNSs to be the main resources for investigation. Examples of LBSNSs are Foursquare, Gowalla, Whrrl and Brightkite. They studied various factors which might dominate the performance of recommendation. These factors comprise geographical, social and temporal influences and they are considered with user’s preferences. In related work, the users’ preferences referred to check-in behaviors of users consisting of users and their checked-in locations. Ye, Yin, and Lee, (2010) began with studying social and geospatial impacts by proposing friend-based CF (FCF) and and Geo-Measured FCF (GMFCF). This work was extended in the next year by Ye, Yin, Lee, and Lee (2011). They proposed a USG framework including user preference, social and geographical influences. The geographical

influence was modeled by adopting a power low distribution based on a Baysian algorithm. The geographical influence was based on an assumption that users tend to check-in POIs close to their living or working places. In case of social influence, they exploited user-based CF for modeling. The results of fusing the two influences with users' preferences revealed that geographical influence provided more important than social influence.

The USG framework proposed by Ye et al. (2011) inspired many researchers to further explore other influences and techniques in order to improve quality of POI recommendation. There were many researchers still conducting geographical and social influences. Cheng, Yang, King and Lyu (2012) introduced the integration between matrix factorization (MF) and geographical and social influences to alleviate a sparsity problem of location recommendation. The notion of social influence was similar to Ye et al. (2011). Nevertheless, they defined geographical influence different from Ye et al. (2011) that users tend to check-in around the famous locations. Therefore, they adopt a multi-center Gaussian model (MGM) to assign the famous locations as centers. Eventually, both influences were integrated with users' preferences by employing probabilistic matrix factorization (PMF) for recommending POIs. iGLR introduced by Zhang and Chow (2013) demonstrated the way to combine the personalized geographic influence with user preference and social influence. The iGLR provided probability which users might visit unseen locations. Picot-Clémente and Bothorel (2013) proposed an algorithm named KatzFSG combining social graph, frequentation graph and geographic graph into one graph. The algorithm has a role in realizing a proximity computation between users and places in the combined graph.

The temporal influence is also studied in the field of POI recommendation. Because of time stamp of check-in history, it exhibits the way to enhance accuracy of recommendation by considering time of a day. Commonly, temporal influence is adopted to predict kinds of locations that users visit in the different time. Rahimi and Wang (2011) conducted temporal and spatial properties with probability distribution function in order to achieve category-based location recommendation. Cheng, Yang, Lyu and King (2012a) investigated the spatial-temporal properties of LBSNS datasets. They proposed a novel MF method named FPMC-LR to incorporate the two properties. Yuan, Cong, Ma, Sun and Magnenat-Thalmann (2013) presented time-aware POI recommendation. They exploited both temporal and spatial influences to improve quality of POI suggestion.

In addition to the study of three influences, some work examined other improvements such as Urban POI-Mine (UPOI-Main) proposed by Ying, Lu, Kuo and Tseng (2012). They extracted the features from social factors, individual preferences and POI popularity in order to build a model using a data mining approach. The model is adopted to recommend POIs that the user may like. Zheng, Jin and Li (2013) proposed a cross-region CF approach for POI recommendation. The approach was applied to recommend POI in a new region for users. Even though the previous attempts had shown the possibility to improve the attraction recommendation systems based on SNSs and LBSNSs, there are some problems appeared in those endeavors, particularly the accuracy of recommendation and other influences.

There are three issues of problem revealed in the existing recommendation system and these issues could be further investigated. The first issue is ignoring incomplete data. Most systems did not mention about incomplete data. There was only one proposed this issue (Xiao et al. 2011). Because of the contents on SNSs and LBSNSs are driven by people, they still have many incomplete contents of attractions. For instance, there are many restaurants on the Facebook labeled their category with "Local Business" (Chatcharaporn, Angskun, and Angskun, 2012). Avoiding these incomplete data causes the system to loss necessary information in analyzing the user preferences. The second issue is generating an off-line filtering model. Data collection from SNSs and LBSNSs in specified periods and regions leads the system to lack up-to-date information for improving inference. The insufficient up-to-date information causes the system cannot infer the current interests of users which may change over time. Besides, there is no mention of using a demographic profile of users for attraction recommendation even though it is a major part of being SNSs. Implementing the SNS profile is a challenge and worth trying for enhancement. The last issue is generating an online filtering model. A large amount of collected data from SNSs and LBSNSs makes the system to consume long processing time in order to build the online filtering model. Moreover, ignoring the measurement of response time makes the users unable to known how long they need to wait for recommendations. Consequently, the response time is one measure raised in this study.

Hence, this research aims to design a mobile engine for personalized tourist attraction recommendation using SNSs. The mobile engine also enables new users to get serendipitous outcomes. Likewise, the mobile engine may enhance tourism

experiences of the tourists by providing the personalized attractions nearby the tourists and navigating them to those destinations favorably.

1.2 Research Objectives

The main objective of this research is to design a mobile engine for personalized tourist attraction recommendation using SNSs that performs the following tasks:

- To revise incomplete contents retrieved from SNSs with the combination of SNSs, Latent Semantic Analysis and Machine Learning technique.
- To create an efficient online filtering technique.
- To improve online response time with a method to decrease the numbers of users who share the mutual interests in the tourism domain.

1.3 Research Hypothesis

1.3.1 The mobile engine can revise the incomplete categories of attractions on SNSs correctly with greater than or equal to 80% of recall.

1.3.2 The mobile engine can provide personalized attractions that match individual travelers' interests correctly with greater than or equal to 80% of recall.

1.3.3 The mobile engine is able to response the user by illustrating feedbacks within 5 seconds.

1.4 Basic Assumption

1.4.1 The active user must have a smart device supported either GPS or wireless network such as 3G and WIFI for detecting the location because the personalized attractions as output of the mobile engine are based on the active user's current location.

1.4.2 The mobile engine has to support Apple iOS and Google Android with Safari and Chrome web browsers.

1.4.3 The mobile engine requires user's permissions to access and retrieve their interests and social contexts in Facebook server.

1.4.4 The mobile engine only supports English language.

1.5 Scope of the Study

The study focuses on designing a mobile engine for personalized tourist attraction recommendation using SNSs. This system has the ability to recommend attractions based on the interests of users retrieved from SNSs including user's demographic and social contexts (e.g., friends and check-in histories). In case of missing category prediction, the process only performs with attractions written in English language. Besides, the mobile engine has capability to recommend the attractions related with the active user's interests in the tourism domain before limiting those attractions locating around the active users. To deliver personalized attractions to the users, the mobile engine demonstrates them on tourists' mobile devices via Me-Locations application.

1.6 Expected Results

1.6.1 Direct Expected Results

1) To achieve a methodology for predicting incomplete category attractions retrieved from the SNSs.

2) To achieve a methodology for creating an efficient online filtering technique.

3) To achieve a methodology for improving online response time with a method to decrease the numbers of users who share the mutual interests in the tourism domain.

1.6.2 Indirect Expected Results

1) To provide a proof of SNSs and LBSNSs that they can enhance the quality of tourism recommendation systems higher than non-personalized recommendation systems.

2) To facilitate the tourists for obtaining personalized attractions and navigate them to those locations.

1.7 Definitions of Terms

Active user is a tourist who is a member of Facebook and has a smart device supported the mobile application named Me-Locations. Furthermore, the active user is required to grant permissions for the application in order to allow the mobile engine to access and fetch his/her SNS information from SNS servers. This action makes the user to have right for deserving personalized recommendations.

Check-in is a social function on LBSNSs in order to share the active user's visiting location to their friends. The capability of current smart devices, particularly GPS function, makes sharing the information on SNSs able to attach with geographical data such as latitude and longitude. These data bring a new dimension to SNSs and it also enables the virtual world to meet the real world. Hence, Check-in function on LBSNSs does not only allow users to declare the places where they are visiting but also lets users to explore sharing places from the others. Checked-in information is maintained in LBSNSs servers and it can be retrieved through API. Eventually, the tourism recommendation system could adopt Check-in to analyze the active users' behaviors in the tourism domain and may lead them to splendid feedbacks.

Demographic Filtering (DF)-based friends are the other users in the mobile engine who have similar demographic information with the active user. The mutuality of demographic between the active user and the others is calculated by adopting a demographic filtering (DF) approach. Examples of demographic profiles are gender, relationship status, living location as well as education information. These data are capable to be extracted from SNS servers. The DF approach uses SNS profiles of the two users to compute similarity between them. A similarity value is the result of the computation. The other users who have the similarity value more than 0 are considered as DF-based friends of the active user.

Electronic tourism (e-Tourism) is using information and communication technology (ICT) to support travel decision making of tourists. The Internet is the most important factor to access the information in order to facilitate the

1decisions such as determining destinations, choosing attractions, selecting accommodations and considering routes.

“Friends” is a term in order to call the other users in the mobile engine who are selected to operate personalized attraction recommendation with the active user. These users can be SNS friends of the active user. Besides, data of all friends as the whole data is adopted to perform data selection. Selecting friends is able to be achieved by using three methods. The first one is using the entire friends. Another one is choosing only SNS friends. The results selected from this method are called SNS friends. Other one is implementing the demographic filtering (DF) approach to reduce the number of friends. The consequences of the last method are called DF-based friends. These friends gotten three data selections finally are taken to analyze the performance of recommendation based on the user-based CF.

Location Based Social Networking Service (LBSNS) is a kind of web services driven by checking-in locations of users. The users operate the checking-in to express others based on their current locations. The current locations are detected from built-in GPS or wireless network in a smart device. These checking-in information of LBSNS can be adopted to imply users’ travelling styles. Besides, the information is able to discover the mutual interests of users based on sharing of similar locations. Due to the plenty of user-location information, LBSNS is implemented with two tasks. The first one is performing the missing category revision. Other one is operating personalized attraction recommendation for tourists.

Mobile application is a web application named “Me-Locations”. It is designed and developed to support presentation on mobile devices. It needs to work with a web browser in order to present the personalized attractions. The application

mainly supports Apple iOS and Google Android. Safari and Chrome are two web browsers recommended to operate Me-Locations on the two mobile operating systems. Using the application enables the active user to view the personalized tourist attractions on a digital map based on his/her current location. Furthermore, the application has competency to navigate the active user to the desired destinations.

Mobile tourism (*m*-Tourism) is an activity of *e*-Tourism performing on mobile devices such as browsing landmark information, booking accommodations or flights, finding attractions as well as navigating routes.

Point of interests (POIs) is specific point locations appeared on the digital map in the form of marks. POIs can provide useful information about places and the users can use the GPS function in order to track to these places.

Smart device is a mobile device built on a mobile computing platform. The smart device can be a phone, a phablet and a tablet. Typically, the smart device such as a smartphone have more capable than a feature phone in data computing and network connecting. Furthermore, the smart device is able to setup applications. This makes the smart device to extend its capabilities in order to respond the need of users. It is a suitable platform for the mobile application.

SNS friends are the users who share the social relationships with the active user in SNSs. The relationships may be established with other users who do not know each other before. After the users created the relationships, they can traverse into these relationships to visit their friends' profiles or interests as well as create a new relationship with others. The tourism recommendation system could exploit the relationships of SNS to discover some users who share similar interests with the active user, especially check-in histories. Furthermore, using SNS friends may

facilitate the mobile engine to limit the numbers of similar users in order to make it faster.

Social Networking Service (SNS) is a kind of web services which store enormous personal data and user-generated contents. In this study, some SNS data are fetched from comprising profiles, social relationships as well as check-in histories. These data are used to identify the active user's interests in the tourism domain and improve the quality of recommendation system. Moreover, SNS profiles of users stored in the mobile engine are also used to seek the similar users who share mutual SNS profiles with the active user.

CHAPTER 2

REVIEW OF THE LITERATURE

The literature review is presented in this chapter comprising the related concepts, theories, as well as background knowledge. This chapter starts with details of text categorization and latent sematic analysis applied for attraction categorization. The second part of this chapter explains personalized recommendation system. Social networking sites are introduced in the third section followed by location-based social networking services. The last but not the least is discussion of related work. The topics in this chapter are as follows.

2.1 Tourist Attraction Categorization

2.1.1 Text Categorization

2.1.2 Latent Semantic Analysis

2.2 Personalized Recommendation System

2.2.1 The Concept of Personalization

2.2.2 Recommendation System

2.2.3 User Modeling

2.2.4 The System Evaluation

2.3 Social Networking Sites

2.3.1 Social Networking Site Definition

2.3.2 Characteristics of Social Network Sites

2.3.3 Social Network Connect Services

2.1 Tourist Attraction Categorization

To overcome incomplete categories of tourist attractions fetched from SNSs, the notions of text categorization and latent semantic analysis (LSA) associate the first objective of this research. Text categorization is one kind of classification tasks in data mining. It has a purpose to assign pre-defined classes to text documents based on their contents. Originally, LSA is as an information retrieval method. However, this technique nowadays has been applied to text categorization in order to improve the computational efficiency. Both text categorization and LSA technique are two significant approaches implemented in the mobile engine in order to label the categories with predicted categories. This section starts with description of text categorization, followed by explanation of LSA.

2.1.1 Text Categorization

The rapid increment of electronic documents on the Internet era causes text categorization to become the key approach for information organization and knowledge discovery. Examples of electronic documents on the Internet are news, blogs, emails and comments. Most of these documents are unstructured and semi

structured contents. Text Classification is raised to deal with those contents. It requires text mining, machine learning and natural language processing techniques to automatically classify and discover patterns from those contents. Generally, text categorization has a main purpose to facilitate users to extract information from texts. Furthermore, it aids users to achieve some operations like retrieval, classification and summarization. Categorizing the tremendous information on the Internet is a very difficult task for human hence the machine plays an important role in solving this problem, especially the capability of automatic text categorization. This subsection presents the detail of text categorization process proposed by Khan, Baharudin, Lee and Khan (2010). Figure 2.1 demonstrates an overview of text categorization process and the detail explanation is based on this overview.

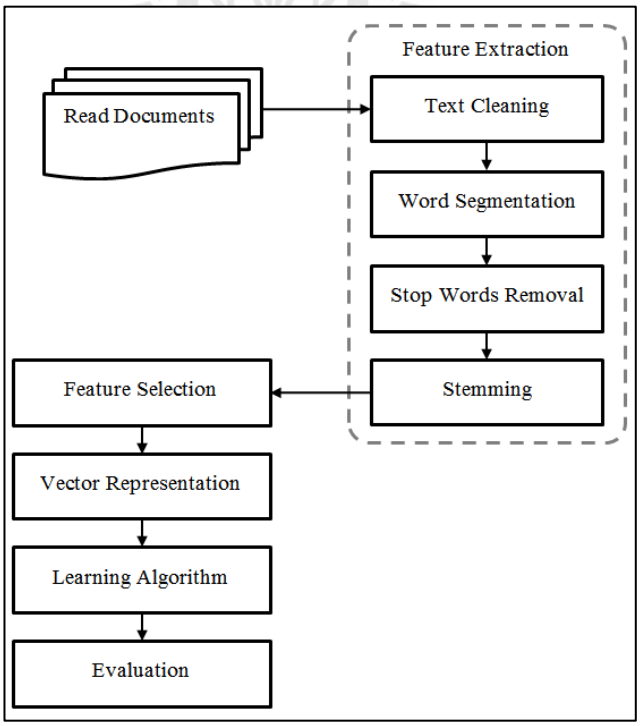


Figure 2.1 An overview of text categorization process.

1) Feature Extraction

The first step of text categorization process is feature extraction. This step is adopted as a pre-processing task in order to eliminate noise from documents and retain the significant words as document representation for training a classifier. Commonly, there are four steps of the feature extraction:

1.1) Text Cleaning: All documents need to be transformed into plain text by removing the symbols and non-alphabetical characters and converting them to a lower case.

1.2) Words Segmentation: The plain text is adopted as a string, then they are separated into a list of words by determining their space as a separator.

1.3) Stop Words Removal: Stop words could be the words that frequently occur in documents or unimportant words, particularly article, preposition, pronoun, and conjunction. These words need to be removed. The examples of stop words comprise “a”, “an”, “the”, “he”, “she”, “they”, “in”, “on”, “and”, “who”, “which”, and “that”.

1.4) Stemming: This step is finding the root form of words by applying the stemming algorithm. The algorithm has capability to convert a contrasting form of words into a similar canonical form. For example, “connects”, “connected”, “connecting”, and “connection” can be stemmed to “connect”.

2) Feature Selection and Feature Transformation

After performing feature extraction, the extracted words are considered to be features. The other crucial step of text categorization is feature selection. The aim of feature selection is to reduce the curse of dimensionality. The

dimensionality reduction is able to improve the scalability, efficiency and accuracy of text categorization. Basically, a good feature selection method should determine domain and algorithm characteristics (Wang, Sun, Zhang and Li, 2006). In order to select subset of features for documents representation, each word needs to be computed with feature evaluation metrics. There are many feature evaluation metrics conducted for feature selection. The well-known metrics are information gain (IG), gain ratio (GR), term frequency (TF), term frequency – inverse document frequency (TF-IDF), and Chi-square (Khan et al., 2010). Selecting these metrics depends on a problem domain and a nature of machine learning algorithms.

a) Term Frequency – Inverse Document Frequency (TF-IDF): TF-IDF is a weighting method extensively used in information retrieval and text categorization. It exploits a statistical measure to evaluate importance of a term in documents. The high weight value indicates that the word has high ability to separate document (Cai, Gokhale and Theiler, 2007). The calculation of TF-IDF can be displayed in Equation 2.1 as follows:

$$w_{(f,d)} = tf_{(f,d)} \times \log \frac{|D|}{|DF_{(f)}|} \tag{2.1}$$

where

$w_{(f,d)}$ is weight of a feature *f* occurs in a document *d*.

$tf_{(f,d)}$ is frequencies of feature *f* which occurs in document *d*.

$|D|$ is the total number of documents in a training set.

$\left|DF_{(f)}\right|$ is the number of documents which a feature *f* appears more than once.

b) Information Gain (IG): IG is one algorithm adopted to select features from training data. The algorithm computes probability of the term in each class of documents (Quinlan, 1993). The term which has a high value of gain shows that it has much effectiveness for classifying the training data. The following equations show computation of IG.

$$IG(t) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + P(t)\sum_{i=1}^{m} P(c_i \mid t)\log P(c_i \mid t) + P(\bar{t})\sum_{i=1}^{m} P(c_i \mid \bar{t})\log P(c_i \mid \bar{t}) \quad (2.2)$$

where

$P(c_i)$ is the probability of a class c_i appearing in the training set.

$P(t)$ is the probability of a term *t* appearing in the training set.

$P(\bar{t})$ is the probability of a term *t* not appearing in the training set.

$P(c_i \mid t)$ is the probability of a class c_i given that the term *t* appears.

$P(c_i \mid \bar{t})$ is the probability of a class c_i given that the term *t* not appears.

c) Gain Ratio (GR): Because the information gain problem about bias has an effect on multidimensional dataset, GR is used to resolve the problem by adjusting information gain (Quinlan, 1993; Hall and Smith, 1998). In text categorization, GR is used to evaluate reliability of terms by computing GR in each

class with the value of information gain. The calculation of GR can be shown in Equation 2.3 as follows:

$$GR(t_k, c_i) = \frac{\sum c \in \{c_i, \bar{c}_i\} \sum t \in \{t_k, \bar{t}_k\} P(t,c) \log \frac{P(t,c)}{P(t)P(c)}}{-\sum c \in \{c_i, \bar{c}_i\} P(c) \log P(c)} \quad (2.3)$$

where $P(t,c)$ is the probability of the term t and class c occurring simultaneously.

d) Chi Square (χ^2): χ^2 is one kind of statistical filtering approaches conducted in the feature selection. This approach has ability to measure the degree of dependency between a term and a specific class (Ying and Pedersen, 1997). This approach can be calculated as:

$$\chi^2(t, c_i) = \frac{D[P(t,c_i)P(\bar{t},\bar{c}_i) - P(t,\bar{c}_i)P(\bar{t},c_i)]^2}{P(t)P(\bar{t})P(c_i)P(\bar{c}_i)} \quad (2.4)$$

where D is the total number of documents.

3) Vector Space Document Representation

Because a document is a sequence of terms, commonly each of documents is corresponded to an array of terms. The relationship between terms and a document is capable to be represented as a vector. Vector of each document is commonly consisted of term weights, where each term is selected from the previous step. Thus, a vector of a document can be defined in Equation 2.5. In case of term

weights, each weight is able to be represented by various assignments such as a binary value, a frequency of terms and a TF-IDF weight.

$$d_j = (w_{j1},...,w_{jn}) \tag{2.5}$$

where w_{ji} is weight of *i*-th indexing terms in the document *j*.

In text categorization, a term-document matrix is a clearly way to demonstrate the relationship between terms and documents. An example of a term-document matrix is illustrated in Table 2.1. As shown in Table 2.1, the rows of the matrix correspond to documents and the columns correspond to terms. Each cell of the matrix consists of binary values used as weights where 0 indicates a term is absent and 1 is otherwise. The last column of the matrix represents the categories of documents. Each document belongs to a category. The category is necessary data for supervised machine learning (ML) algorithms. It can be exploited by ML algorithms in order to discover the patterns of each category. Hence, the following step of text categorization conducts the document vectors attached with category displayed in Equation 2.6 as inputs for constructing the categorization models with ML algorithms.

$$d_j = (w_{j1},...,w_{jn},\#category) \tag{2.6}$$

Table 2.1 An example of term-document matrix representing the relationship between terms and document based on a binary value.

Documents	**Terms**				**Categories**
	$Term_1$	**$Term_2$**	**…**	**$Term_j$**	
Doc_1	0	1	…	0	Spam
Doc_2	1	1	…	0	Not Spam
Doc_3	0	0	…	1	Not Spam
…	…	…	…	…	…
Doc_i	0	1	…	1	Spam

4) Machine Learning Algorithms

Electronic documents are able to be categorized by an unsupervised method (i.e., clustering) and a supervised method (i.e., classification). This study mainly focuses on the supervised classification technique. The task of automatic text categorization tends to be implemented by ML approaches such as Decision Tree, Bayesian classifier, Neural Networks, and Support Vector Machine (SVM). The relationship between terms and documents in the form of vectors can be used by these ML algorithms. The vectors are exploited as a training set for model construction. The categorization model has a role in assigning pre-defined categories to unseen documents by considering the similarity of content between unseen documents and the training set. There are four kinds of ML algorithm proposed in this work and their details can be explained as follows.

a) Decision Tree

The decision tree re-constructs the categories of training set by creating true/false conditions in the form of a tree structure. The structure of decision tree has leaves or nodes representing the categories of documents and branches representing the rules leading to those categories. The well-organized decision tree can simply categorize a document by inserting the document in the root node of the tree, then let it traverse through the query structure until it reaches a particular leaf. The particular leaf is considered as the category of the document.

The decision tree is an outstanding ML algorithm for text categorization due to its simplicity. The simplicity is the main advantage of this algorithm because the illustration of decision rules in the structure of tree is easily understanding and interpreting for users. The well-known decision tree algorithms include ID3 and J48. Nevertheless, the algorithm is able to provide poor quality of categorization if it is implemented with the large number of features. Building a decision tree categorization model with a numerous number of dataset leads a complex and large structure of tree. Furthermore, this incidence could lead to an over-fitting problem. This problem makes the model to solely deliver the great categorizing result with training and test sets.

b) Naïve Bayes Algorithm

Naïve Bayes classifier exploits probability to estimate the categories of documents based on Bayes' Theorem with strong independence assumptions. The probability directly corresponds to the independent feature model. The independence assumptions of features cause the order of features to be irrelevant.

Therefore, a feature does not affect to other features in classification tasks (Brücher, Knolmayer and Mittermayer, 2002). These assumptions lead an operation of Bayesian approach to be more productive. However, the assumption could restrict its implementation. With the nature of probability model, a categorization model based on naïve Bayes is able to be trained very capably because it can be implemented with a small amount of training data in order to specify the essential parameters for categorization.

As mentioned above, demanding a small amount of training data for model construction is the strength of naïve Bayes algorithm. In addition, the algorithm has capability to provide the correct category with more probable than others. Even though category's probabilities do not have to be approximated very well, the overall categorization model is robust enough to neglect crucial insufficiencies in its underlying naïve probability model. Nonetheless, the main shortcoming of naïve Bayes algorithm is offering low performance of categorization when compared with other algorithms such as Support Vector Machine and Artificial Neural Network. Equation 2.7 and 2.8 explain the calculation of naïve Bayes algorithm for document categorization.

$$P(c_i \mid D) = \frac{P(c_i) \times P(D \mid c_i)}{P(D)} \tag{2.7}$$

$$P(D \mid c_i) = \prod_{j=1}^{n} P(d_j \mid c_i) \tag{2.8}$$

where

$P(c_i \mid D)$ is the probability that the documents belong to the category c_i.

$P(c_i)$ is the probability of a given category c_i.

$P(D)$ is the probability of the documents.

$P(D \mid c_i)$ is the probability that the documents are in category c_i.

c) Artificial Neural Network

Artificial neural network is an interconnected group of nodes. These nodes are also called artificial neuron. The neurons are interconnected into a group using a mathematical model for information processing. The neural networks make their neuron sensitive to store an item. It is capable to be adopted for distortion tolerant storing of a large number of cases represented by high dimensional vectors.

There are various approaches of neural network implemented in document categorization tasks. Some researchers applied the single-layer-perceptron consisting of one input layer and one output layer (Ng, Goh and Low, 1997). The inputs are directly fed to the outputs through a series of weights as demonstrated in Figure 2.2. This is the simplest way of feed-forward network. Adopting multi-layer perceptron is more sophisticated and widely implemented in the categorization task (Ruiz and Srinivasan, 1998). The approach comprises an input layer, one or more hidden layers, and an output layer.

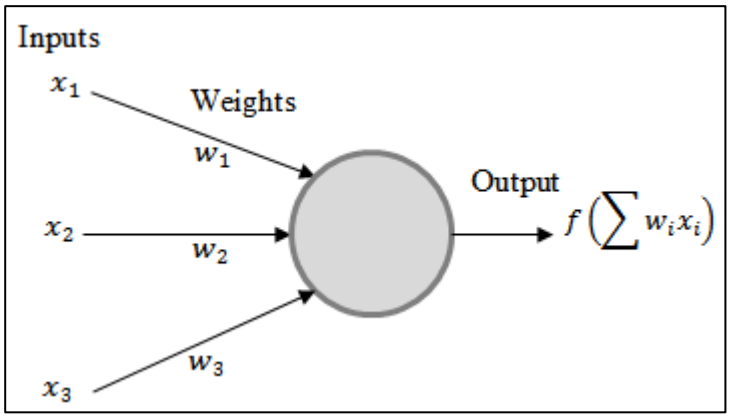


Figure 2.2 Artificial neural networks with the single-layer perceptron.

The advantage of artificial neural network for text categorization is competency of handling documents with high dimensional features. Besides, it can be utilized to documents with noisy and contradictory data. It also provides linear speeding up in matching a process when operating with the large number of computational nodes. This improvement is provided by a parallel computing architecture, where each node can compare its input value against the value of stored cases from others independently (Myllymaki and Tirri, 1993). In case of disadvantage, artificial neural network requires high computing cost, especially CPU and physical memory usage. Another case of disadvantage is that the artificial neural networks are very difficult to understand for common users.

Enhancing the performance of text categorization systems based on the neural network algorithm has been proposed in recent years. Back-propagation neural network (BPNN) and modified back-propagation neural network (MBPNN) for text categorization models are introduced by Yu, Xu, and Li (2008). These researchers also proposed a latent semantic analysis (LSA) as an efficient feature

selection method in order to reduce the dimensionality and improve performance of categorization models.

d) Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of efficient approaches for text categorization because it is able to exhibit the good results. The categorization method of SVM is based on the Structural Risk Minimization which is principle from computational learning theory (Vapnik, 1995). The notion of the principle is to seek a hypothesis in order to confirm the minimum error. Basically, SVM requires both positive and negative training sets. The requirement is different from other ML algorithms. The positive and negative training sets are adopted to discover the decision hyperplane in order to separate the two sets with the maximum margin. The training documents which are closest to the decision hyperplane are determined to be the support vectors. The competency of SVM categorization is unchanged if the documents which do not belong to the support vectors are eliminated from the training set (Joachims, 1998).

The strong point of SVM is delivering performance of categorization with more efficient than other approaches. Moreover, the technique has competency to cope documents with high-dimensional spaces. Nevertheless, the main drawback of SVM is the complicated algorithms for training and categorizing data. The drawback causes the machine to consume high computational resources both CPU and memory when it needs to construct the categorization model. Furthermore, in case of using the documents associated with several categories, this action could lead to the confusions in the categorization task because SVM typically

calculates the similarity between documents for each category separately (Brücher et al., 2002).

5) Evaluation

The evaluation of text categorization has an objective to measure the effectiveness of categorization model. Normally, the dataset is divided into two sets called training set and test set. The training set is manipulated to train itself with ML algorithms for model construction. The test set is conducted to test the performance of the model. Even though there are many various measures for performance evaluation, the most often used measures are precision, recall, and f-measure. The three measures come from information retrieval science. A confusion matrix is proposed to explain the way to compute the three measures. As illustrated in Table 2.2, there are four values appeared in the confusion matrix including true positive (*TP*), false negative (*FP*), true negative (*TN*), and false negative (*FN*). *TP* represents the number of documents correctly categorized. *FN* corresponds to the number of documents incorrectly categorized. *FP* is the number of documents that are not labeled to the particular category but should be. *TN* expresses the number of documents classified as negative and its actual category is negative as well. The calculation of the three evaluation metrics is exhibited in Equation 2.9, 2.10 and 2.11 (Sokolova and Lapalme, 2009), respectively.

Table 2.2 Confusion matrix represented evaluating efficiency of categorization model based on information retrieval measurements.

Classified Category	**Actual Category**	
	Positive	**Negative**
Positive	*TP*	*FP*
Negative	*FN*	*TN*

$$Precision = \frac{TP}{TP + FP} \tag{2.9}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.10}$$

$$F - measure = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{2.11}$$

This study exploits the capability of text categorization to revise the incomplete categories of attractions collected from SNSs. This approach is performed with the title of attraction in order to label the predicted categories. To find the best solution for categorizing attractions, there is a performance comparison of categorization models. Those models are generated with various ML algorithms including Naïve Bayes (NB), Decision Tree (J48), Back Propagation Neuron Network (BPNN), and Support Vector Machine (SVM).

2.1.2 Latent Semantic Analysis

Latent semantic analysis (LSA) is a well-known technique adopted in text categorization and information retrieval. The technique was proposed by Deerwester, Demais, Furnas, Landauer and Harshman (1990). LSA exploits statistics and linear algebra in order to capture and extract the semantic content from text (Lv and Lui, 2005). The semantic content corresponds to the meaning of words which is calculated by using statistical methods. This makes texts be able to compare the similarity between them by using the semantic content. Even though many work demonstrated the different number of LSA steps, there are three common steps of LSA-based information retrieval as follows.

1) Preparing Term-Document Matrix

The first common step is preparing a term-document matrix. The term-document matrix $A(m \times n)$ is generated to represent the significant of terms or words in all documents. This means that there are m unique terms in n documents, where $m \geq n$. The unique terms exclude stop words and terms that seldom occur in order to reduce computational burden. The stop words are words in article, preposition, pronoun, and conjunction, such as “a”, “an”, “the”, “he”, “she”, “they”, “in”, “on”, “and”, “who”, “which”, and “that”. Each cell of matrix A_{ij} consists of weights of terms in documents. These weights are used to indicate the importance of a term in a document.

There are many methods adopted to assign the weights. Binary values can be used to represent the weights where 1 corresponds to a term occurred in a particular document and 0 is otherwise. Other simple way is using a term frequency (TF) method. This method employs the number of times that a term i occurs in a

document *j*. Nevertheless, TF method has some disadvantages, particularly in case of long documents. Frequency of terms in these documents can be high which leads to bias. To prevent the bias, a term frequency – inverse document frequency (TF-IDF) method is proposed. TF-IDF assesses the significant of a word in a document by considering both the number of times which a term appears in the document, and frequency of a term in the entire document. The high value of TF-IDF weight is gotten from a high term frequency and a low document frequency of the term in the whole document. The computational method of TF-IDF is explained in Equation 2.1.

2) Performing Singular Value Decomposition

The second common step of LSA process is performing singular value decomposition (SVD). SVD has a role in projecting the term-document matrix *A* onto a latent semantic space (Landauer, Foltz, and Laham, 1998). It manipulates linear algebra to compute the singular value. The singular value decomposition of matrix *A* implementing the SVD is defined as Equation 2.12.

$$A = USV^T \tag{2.12}$$

where U and V^T are the orthogonal matrices of the term and document vectors, and S is the diagonal matrix of singular values.

The SVD computation consists of finding the eigenvalues and eigenvectors of AA^T and A^TA. The eigenvectors of A^TA make up the columns of matrix V, the eigenvectors of AA^T make up the columns of matrix U. The singular

values in matrix S are square roots of eigenvalues obtained from AA^T or A^TA. Figure 2.3 presents decomposition of the matrix A into three matrices.

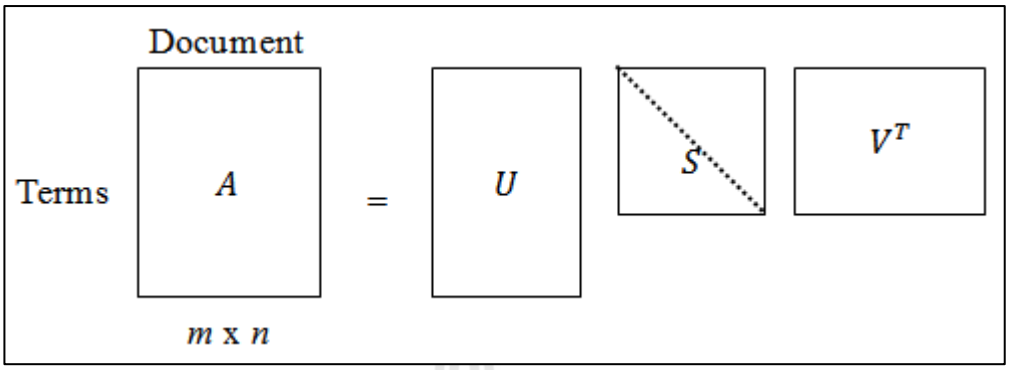


Figure 2.3 The decomposition of term-document matrix A into three matrices U, S and V.

3) Reducing Latent Semantic Space

After computing SVD is finished, the last common step is reducing latent semantic space. In this step, the appropriate k value of matrix S is selected to reduce the feature space because the diagonal elements of matrix S are ordered from most to least significant. Hence, matrix U and matrix V should be truncated as well. The objectives of k value selection are to remove noise from the semantic space and decrease the computational resources, especially the memory usage (Huang, 2011). The results of dimensionality reduction of U, S, V^T are $m \times k$, $k \times k$, and $k \times n$, respectively. Therefore, the approximation of A with rank-k matrix is given by Equation 2.13.

$$A \approx \hat{A} = U_k S_k V_k^T \tag{2.13}$$

where

U_k is composed of the first *k* columns of the matrix *U*.

V_k^T is composed of the first *k* rows of the matrix V^T.

S_k is the first *k* diagonal factors of matrix *S*.

The matrix $\hat{A}$ is considered to capture the most significant relationship between terms and documents.

The result of dimensionality reduction is illustrated in Figure 2.4. The implementation of the three reduced matrices $U_k S_k V_k^T$ obtained from LSA depends on the purpose of applications. The following implementations illustrate the ways to adopt the three matrices for information retrieval and text categorization.

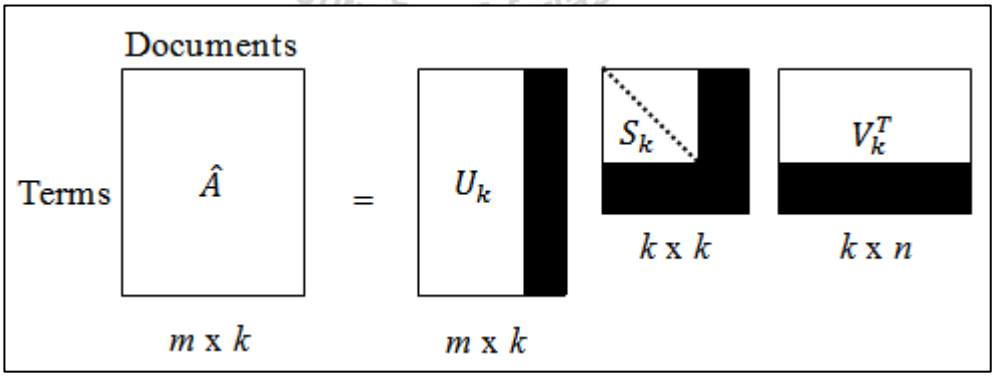


Figure 2.4 Choosing the *k*-largest singular values of three matrices.

4) LSA for Text Categorization

Originally, LSA is proposed for information retrieval and text categorization. A query needs to be represented as a vector with *k*-dimensional latent sematic space as same as the document collection. Then the query is utilized to measure the similarity with the documents. The query is represented by Equation 2.14.

$$\hat{q} = q^T U_k S_k^{-1} \tag{2.14}$$

And each of documents is represented by Equation 2.15.

$$\hat{d} = d^T U_k S_k^{-1} \tag{2.15}$$

The vectors of the query and documents are able to be adopted to compute the similarity between them by using vector-based similarity measures such Cosine similarity (Wang and Zheng, 2013). Equation 2.16 demonstrates the calculation of Cosine similarity.

$$sim(\hat{d}_j, \hat{q}) = \frac{\hat{d}_j \cdot \hat{q}}{\left\|\hat{d}_j\right\| \times \left\|\hat{q}\right\|} \tag{2.16}$$

where

$\hat{d}_j \cdot \hat{q}$ represents the dot product between the sematic vector document $\hat{d}_j$ and the semantic vector of the query $\hat{q}$.

$\left\|\hat{d}_j\right\|$ represents the length of sematic vector $\hat{d}_j$.

$\|\hat{q}\|$ represents the length of sematic vector $\hat{q}$.

With Cosine similarity approach, the similarity score ranges from 0 to 1 where 1 indicates perfect similarity and 0 indicates they are not similar. The category of the query could be considered as the category of the most related document using the score. Nevertheless, implementing LSA with machine learning (ML) is slightly different. In this case, the sematic vectors of document are required to be mapped with their original category. The category tends to be assigned the last value of each sematic vector as shown in Equation 2.17.

$$\hat{d}_j = \left[w_{j1}, w_{j2}, ..., w_{jn}, \#category\right] \tag{2.17}$$

where

w_{jn} is weight of *i*-th indexing terms in the sematic vector document $\hat{d}_j$.

n is the total number of indexing terms.

#category is a original category of the document *d*.

The sematic vectors of all documents are capable to be adopted as input of ML algorithms. ML exploits the input to learn and construct the categorization model in order to predict category of new documents. The famous ML

algorithms in the text categorization task are Naïve Bayes, Decision Tree, Neural Networks, and Support Vector Machine.

In this study, LSA plays an important role in capturing the most important relationship between words and titles of attractions. Furthermore, the latent semantic space as the output of LSA will be conducted as the input of ML technique. Hence, in the text categorization process, LSA technique is implemented after feature selection process. The main benefits of LSA are removing noise from the data and reducing dimension of feature space. The dimensionality reduction alleviates the computational cost of ML, particularly the memory usage when it needs to construct the categorization models. Finally, these benefits of LSA are demonstrated to enhance both quality of categorization and processing time of model construction. However, latent semantic feature selection needs to be investigated, particularly the number of features. Thus, in order to find the appropriate number of features, various sizes of dimensions are performed with model construction of text categorization and evaluated their performance.

2.2 Personalized Recommendation System

In order to recommend personalized attractions for tourists, concepts and theories of personalization directly relate to this research objective. The personalization is an approach of user's interest acquisition in order to create a user model. The user model is a necessary component adopted in recommendation systems. Most of recommendation systems need to maintain the user model in order

to predict users' preferences for suggestion. Hence, this section explains the detail of personalization and recommendation system.

2.2.1 The Concepts of Personalization

Recently, the term personalization is extensively used. According to Kim (2002), main concept of personalization is obtaining some information from the whole information. The information must associate with interests of an individual or a group of individual. For instance, a customer only subscribes sports, news and comedy series from many hundreds of television programs for three months. Furthermore, the term is able to support the idea of one-to-one marketing. Both traditional commercial and e-Commerce exploit one-to-one marketing as business strategy to achieve personalization with their customers. For example, offering products or services in what customers may be interested. Another example is identifying target customers for new products from existing customer bases.

In the first context, some part of the whole information will be delivered to response requirement of a person or a group of persons. The information transfer also depends on requests of users in specific time and format. Some information may not need up-to-date information, e.g., last year a report can be used. The delivered information may obtain from a single source or several sources.

Personalization in the second context comes from the concept of one-to-one marketing. In this concept, the business only needs to operate marketing with some groups of customers. It is used to increase revenue and decrease loss of business opportunity. Hence, it is necessary for business to understand the demand of customers such as habits, lifestyles, preferences as well as likes or dislikes of

products and services. One-to-one marketing makes the business to execute marketing strategy different from its competitors. Furthermore, it assists the business to specify prices of products and plan strategies for each group of customers. This makes business to be more successful to obtain new customers, sustain existent customers and sell additional products or services to the existence. In aspect of customers, they desire a just-in-time recommendation and useful information in order to purchase products or services matching to their interests.

One-to-one marketing is not a novel concept but it has been used since non e-commerce era. In the past, if business owners could recognize their customers, they might know customers' background from conversations. Consequently, when there were new arrival products, the owners knew which customers could be suggested for these products. Another example is the waiter might remember regular customers and could recommend dishes in which customers may be interested.

Although the two contexts of personalization are difference, the mutual between them is a group of individuals who shares common interests and characteristics. The group can be one individual or a group of individual.

Both contexts of personalization involve delivering personalized attractions to tourists in this study. According to the first context, only some attractions should be retrieved and delivered to the users based on their interests on SNSs. Furthermore, another context demonstrates the idea of understanding individual requirement and background of travelers before suggesting attractions to them.

2.2.2 Recommendation System

Originally, recommendation systems are defined as systems which their inputs are recommendations from people. Then, the system gathers those inputs and leads them to proper users (Resnick and Varian, 1997, p.56). According to Burke (2002), he described that "the recommendation system is a system, which produces individualized recommendations as output or having the effect of guiding user in a personalized manner to interesting or useful objects in a large space of possible options". There are numerous researchers which acknowledge Burke's definition. They commented that the recommendation system is a system which can create, search and suggest data such as products or services based on interests of an individual (Melville and Sindhwani, 2010, p.829; Kurashima et al., 2010, p.579). Burke also stated that the recommendation system is adopted to suggest contents from enormous data source such as the Internet. Currently, recommendation systems have been used in famous e-commerce sites such as Amazone.com, eBay as well as CDNow. The purpose is to recommend products and services based on each customer's interests. Therefore, recommendation systems on those sites require customer data such as demographic data and purchase history in order to predict their buying behavior in the future (Schafer, Konstan, and Riedl, 1999, p.158; Burke, 2002; Kabassi 2010).

The perspective of recommendation system in this study is based on the definition of Burke (2002). Because suggesting attractions to tourists is relied on the individual interests of them, only some attractions will be extracted from the entire dataset to be recommended as personalized attractions.

The following contents are detail of recommendation techniques. However, this study emphasizes two filtering techniques named collaborative filtering and demographic filtering. Both techniques are investigated their performance in the aspect of personalized attractions recommendation.

1) Content-based Filtering Techniques

In a recommendation system based on content-based filtering (CBF), products and services will be delivered to the users based on their purchasing or searching behaviors in the past. These purchased products and services have description for comparison with others. Information retrieval technique such as feature extraction is adopted to extract words of those descriptions as features for comparison. TripleHops, TripMatchet2 and Vacation are examples of recommendation system in tourism domain applied with CBF techniques. These systems store user's preferences in its database in order to match with other destinations (Kabassi, 2010).

Some attempts developed CBF technique with simple algorithms in order to collate the features of products and services with user's historical data. Each algorithm has capability of prediction in various ways. The famous algorithms used for CBF techniques are decision tree, neural networks, Bayesian networks as well as rule-based reasoning. Those algorithms have both advantage and disadvantage (Kabassi, 2010).

a) Advantage of Content-based Filtering Techniques

- CBF techniques are directly based on the facts of users, thus the system exploits these facts to achieve the recommendation.

- User model of CBF techniques comes from storing preferences or behaviors of users from the beginning. Thus in the long term, the system can track changing of user's behaviors.

b) Disadvantage of Content-based Filtering Techniques

- Due to the fact that the prediction is based on user's preference, thus the system could provide results that user may already know (Montaner et al., 2003).

- In some cases, these techniques could not suggest satisfied feedbacks. For instance, the user would like to search some trips for her friend but she got a list of trips matching with her individual interests instead.

However, the problem occurs with the recommendation system for tourism infrequently. Most of tourists tend to select services related to their needs instead of others. Nonetheless, this case may happen with recommendation systems for books or movies. The obvious drawback of CBF is a cold-start problem. The problem takes place from the beginning of usage due to lack of user's preferences. This causes the system to provide the feedbacks with low accuracy (Montaner et al., 2003). Hence to overcome this shortcoming, CBF should be co-operated with other filtering techniques such as stereotypes/demographic or collaborative (Rich, 1989; Rich, 1999).

2) Collaborative Filtering Techniques

Collaborative filtering (CF) is a well-known approach widely adopted in recommendation systems. Techniques of CF focus on mutual interests and

behavior of users in the system to execute suggestions. CF differs from CBF whereas CBF is relied on similarity of item's description. On the other hand, CF performs recommendation by matching the active user with others who share common interests. Commonly, CF systems are operated by gathering the user preferences in the form of rating for items in a given domain (e.g., books, movies, music, articles and attractions). The systems exploit the rating behavior of the active user to find similarity between other users in order to recommend particular items. This means that the active user will be suggested a list of items rated by other users. Hence, the CF has ability to advise information for new active users based on their neighbors. The techniques are also applied to reduce the limitation of CBF. The recommendation systems for tourism such as MAIS and TripleHops adopted this approach to recommend some trips to the users (Kabassi, 2010). The system such as TripleHops took statistical data of the active user in the past applied with CBF in order to calculate weight of similarity between the active user and neighbors. The following explanation of CF process is based on view of Sarwar, Karypis, Konstan, and Riedl (2001). Sarwar et al. (2001) proposed the user-based CF relied on a memory-based algorithm. This study raised the user-based CF for implementation because it has performed well for the attraction recommendation (Ye, et al. 2011). The procedure of the user-based CF has four steps as shown in Figure 2.5.

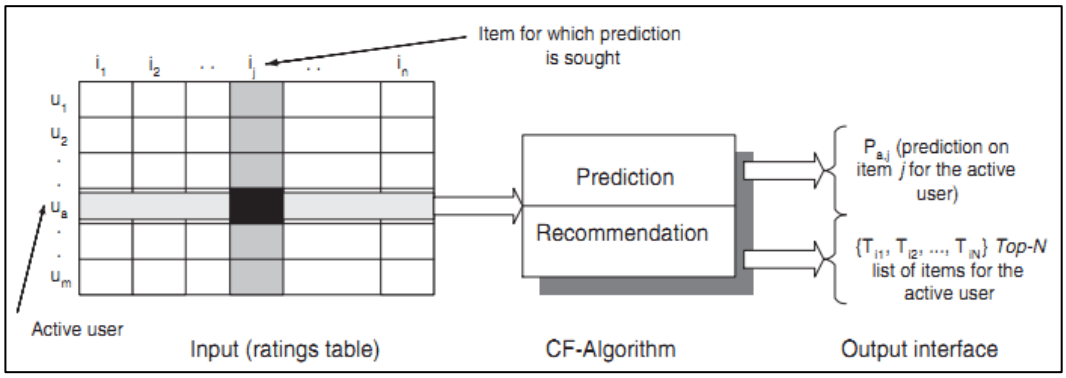


Figure 2.5 An overview of CF process (Sarwar et al., 2001).

2.1) Input Ratings Table

As mentioned above, CF techniques commonly take ratings of users to achieve recommendation. Basically, there are various forms of ratings adopted in a CF recommendation system (Schafer et al., 2007) as follows.

- Scalar ratings can be represented in the form of either numerical ratings range 1-5 or ordinal ratings such as "strongly agree, agree, neutral, disagree, and strongly disagree".
- Binary ratings can be indicated by two choices between agree/disagree, good/bad or checked-in/not check-in.
- Unary ratings can identify that a user has interacted an item. The clear examples of unary ratings are "Like" and "Check-in" buttons of Facebook. The absence of a rating illustrates that there is no information associating between the user and the item.

In addition to types of ratings, acquiring ratings is important for CF recommendation. Rating acquisition is divided into explicit and implicit ratings.

- Explicit ratings are an obvious way achieved by asking an opinion of user on an item. With this approach, initially the user is required to answer preliminary questions about his/her preferences or background.

- Implicit ratings adopt an opposite way to obtain user ratings. The technique observes behavior of users in the system to make the inference for the user preferences. For instance, a user who purchases a product should have stronger interests that a user who just visits a webpage of the same product.

Back to the first stage of CF process, each user assigns ratings to a number of attractions. These ratings of user represent profile or preferences of the user. As shown in Figure 2.6, there are examples of user-attraction matrix. The first one is using an explicit method to acquire user ratings. The scalar ratings are user ratings which have range 1 to 5 of five users for five attractions where 1 is dislike and 5 is the most favorite. Other one is adopting an implicit method to infer user ratings representing 1 for an attraction checked-in by a user while 0 for an attraction unchecked-in by a user. In Figure 2.6 (left), the empty cells of user-attraction matrix mean the attractions are not rated from the users.

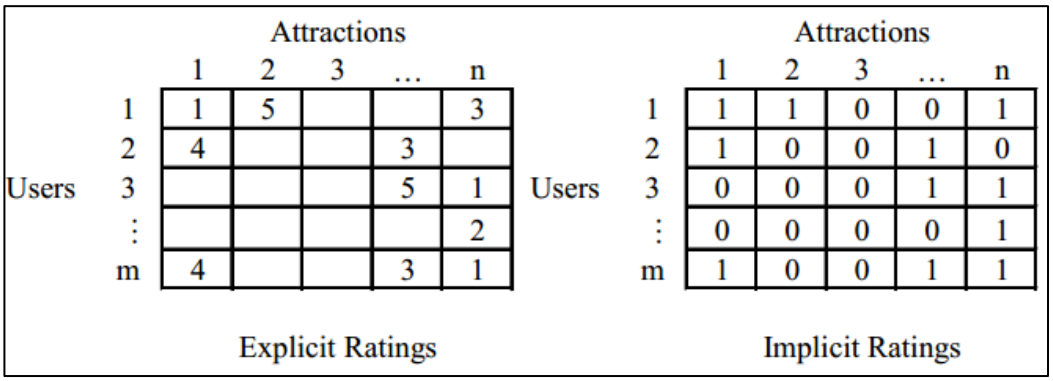


Figure 2.6 Examples of user-attraction matrices adopted in a CF technique based on explicit ratings (left), or based on implicit ratings (right).

Similarity computations as the following process are explained based on both ratings acquisition methods. Nevertheless, this study selects the implicit ratings with binary values to represent the relationship between user and attraction where 1 indicates that a user has checked-in an attraction while 0 is otherwise. In case of implicit ratings, it is raised to infer user ratings based on those binary values. For example, although a user have checked-in Siam Paragon twenty times, the system deduces that a rating of the user on Siam Paragon is 1.

2.2) Similarity Computations

The most crucial process of user-based CF is computing similarity between users in order to acquire the neighbors. The recommendation system needs to consider the active user with the others in the rating matrix. The assumption of finding similarity is that the users who have similar preferences tend to rate the same items. As exhibited in Figure 2.7, it observes that the first user has

similar taste with the second user because they rate on the same attraction. The attraction number one has been rated with the two users called “Co-Rated” item. The system determines the co-rated items to find the similarity between the two users. In order to find the similarity among the users, there are many similarity measures proposed. The well-known similarity measures of CF are Pearson correlation, Cosine similarity and Jaccard coefficient. The following examples are similarity calculation between the active user *u* and the user *v* depended on the mentioned measures.

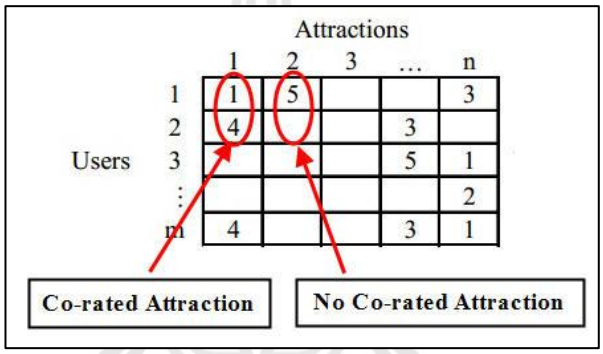


Figure 2.7 Examples of co-rated and no co-rated attractions.

Pearson correlation measures the linear relationship between two variables. The similarity value of Pearson correlation ranges from -1 to 1, where -1 indicates perfect disagreement while 1 indicates perfect agreement. Equation 2.18 defines the calculation of Pearson correlation.

$$pearson(u,v) = \frac{\sum_{k=1}^{n}(r_{u,k} - \bar{r}_u)(r_{v,k} - \bar{r}_v)}{\sqrt{\sum_{k=1}^{n}(r_{u,k} - \bar{r}_u)^2}\sqrt{\sum_{k=1}^{n}(r_{v,k} - \bar{r}_v)^2}} \quad (2.18)$$

where

$pearson(u,v)$ is a value of similarity between the active user *u* and the user *v*.

$r_{u,k}$ is a rating score of the active user *u* to attraction *k*.

$r_{v,k}$ is a rating score of the user *v* to attraction *k*.

$\bar{r}_u$ is an average rating score of the active user *u* to total attraction rated by the active user *u*.

$\bar{r}_v$ is an average rating score of the user *v* to total attraction rated by the user *v*.

n is a number of co-rated items between two users.

Cosine similarity conducts the ratings of the active user *u* and the user *v* to be vectors with *n*-dimensional space. The result of Cosine similarity is between 0 and 1, where 0 indicates two users are not similar and 1 indicates perfect similarity. The computation of Cosine similarity can be defined as Equation 2.19.

$$cosine(u,v) = \frac{u \cdot v}{\|u\| \times \|v\|} = \frac{\sum_{k=1}^{n} r_{u,k} r_{v,k}}{\sqrt{\sum_{k=1}^{n} r_{u,k}^{\ 2}} \sqrt{\sum_{k=1}^{n} r_{v,k}^{\ 2}}} \tag{2.19}$$

where $u \cdot v$ represents the dot product between vector *u* and vector *v*, $\|u\|$ represents the length of vector *u*, and *n* is a number of attractions rated by both users.

Jaccard coefficient is a measure adopted to calculate the similarity between two users with binary variables. It uses two sets of attractions rated by the two users for computation. The similarity is defined as the size of the

intersection divided by the union of the two sets of attractions. The range of Jaccard similarity is between 0 and 1, where 0 indicates two users are not similar and 1 indicates perfect similarity. The calculation of Jaccard coefficient is displayed in Equation 2.20.

$$jaccard(u,v) = \frac{|u \cap v|}{|u \cup v|} \tag{2.20}$$

where u and v are the sets of attractions rated by the active user u and the user v, and $jaccard(u,v)$ is the result of the two sets' intersection divided by their union.

The consequence gotten from these similarity measures indicates the neighborhood between the active user and others in the system. Selecting neighbors for prediction execution can be done either by picking a given number of most similar users (k-nearest neighbors) or choosing all neighbors within a given threshold of similarity. The similarity value is used as weight in the final process of CF for computation. The final process is to provide some personalized items based on individual interests of the active user. There are two kinds of offering the results relied on the objective of recommendation systems. The first one is prediction operating to predict rating of the particular item that the active user has not rated before the user. The second one is recommendation producing a top-N recommendation to the active user. The top-N recommendation is a list of unrated items which are relevant to the active user's preferences.

2.3) Prediction

After calculation of similar neighbors is finished, prediction process takes *k*-nearest neighbors to produce the predicted rating for the active user. The similarity values between the users are used as weights. Besides, with different similarity measures, the formula of prediction is distinct. Equation 2.21 shows the way of performing prediction with using scalar ratings. Without the average rating of the active user, binary ratings tend to be implemented with top-*N* recommendation.

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n=1}^{N} W_{u,n}(r_{n,i} - \bar{r}_n)}{\sum_{n=1}^{N} W_{u,n}} \tag{2.21}$$

where

$pred(u,i)$ is predicted rating of attraction *i* for the active user *u*.

$\bar{r}_u$ is the mean of rating given by the active user *u*.

$W_{u,n}$ is a weight reflected the similarity between the active user *u* and the neighbor *n*.

$r_{n,i}$ is a rating score of the neighbor *n* to attraction *i*.

$\bar{r}_n$ is the mean of rating given by the neighbor *n*.

N is the total number of nearest neighbors.

2.4) Recommendation

The *k*-nearest neighbors are also adopted in this recommendation step. Typically, the recommendation systems select the top-*N* attractions by

considering the predicted ratings. However in case of binary variables, it determines the following score proposed by Weiss and Indurkhya (2001):

$$score(u,i) = \sum_{n=1}^{N} W_{u,n} r_{n,i} \tag{2.22}$$

where $score(u,i)$ is the score of unrated attraction i for the active user u, $W_{u,n}$ is the similarity between the active user u and the neighbor n based on Jaccard or Cosine similarity measures, $r_{n,i}$ is the rating of the neighbor n on attraction i, and N is set of the nearest neighbors.

Without the normalization, the score is easily adopted to select the top-N attractions with highest score. Selecting value of N depends on the system developer to pick the appropriate number of results before demonstrating them to the active user.

a) Advantage of Collaborative Filtering Techniques

- The approach can solve the drawback of content-based approach by using other users for recommendation instead of considering description of items.

- CF is capable to provide recommendation based on either user or item. The user-based CF suggests unseen items already rated by similar users. Another one advises items including the high correlation based on previous rated items of the active user. This capability make the system based on CF is able to suggest serendipitous outcomes (Montaner et al., 2003).

b) Disadvantage of Collaborative Filtering Techniques

There are three main drawbacks of CF including sparsity, cold-start and scalability. The details of these drawbacks can be described as follows:

- Sparsity tends to occur with a large system when a number of items are larger than a number of users. This reduces a chance of rating at the same item of users. The incident causes rating of users to become too sparse and recommendation system to work harder for seeking the similar users. Hence, the problem leads the system to provide low accuracy feedbacks.

- Cold-start can happen when there are new items or users arrived in the system. Because the new items have not been rated by any users, the system cannot allow these items to recommend to the users. Another case is new users. If they did not rate any items, the system could not discover the similar neighbor and operate recommendation. Even though the system could perform suggestion, the user would be suffered from low accuracy outcomes.

- Basically, recommendation systems with CF approach, especially memory-based CF, tend to consume the amount of memory and processing time linearly with increasing the number of users and items. Hence, insufficient resources can lead to the scalability problem.

In this study, user-based CF plays a significant role in recommending personalized attractions. Implicit ratings and binary variables are considered to represent the preferences of the users in the mobile engine. Both Cosine and Jaccard similarities are the two measures for finding the *k*-nearest neighbors.

Recommendation is the final step of CF in this work manipulated to provide top-*N* recommendation for the active user.

3) Demographic Filtering Techniques

Demographic filtering (DF) techniques exploit descriptions of people to learn the association between certain items and the types of people who like it (Pazzani, 1999). The user model of DF-based system is generated by classifying users based on their personal attributes and adopted to provide recommendation based on the demographic classes. The examples of personal attributes include age, gender, education, geographical location, etc. Typically, personal data is obtained from the system registration. These data will be adopted to construct the user model by using the machine learning technique such as classification and clustering. Nonetheless, the notion of DF is similar to CF, especially comparing users against each other but CF uses ratings instead of demographic data (Burke, 2002). Pazzani (1999) took DF with machine learning to apply with her recommendation system by extracting demographical data from user's home pages in order to predict favorite restaurants of users. Traveller (Schiaffino and Amandi, 2009) adopted DF with user profiles by comparing each attribute of user profile with the corresponding attribute of the package tour before suggesting a package tour to the user.

The following explanation shows the method of using demographic data in order to find the similar users. The explanation is followed the perspective of Vozalis and Margaritis (2004) which referenced the original concept introduced by Pazzani (1999).

Initially, DF-based recommendation systems need to select demographic attributes of users in order to form a user-demographic matrix. Table 2.3 demonstrates sample of selected demographic attributes for the user-demographic matrix construction. Furthermore, Table 2.3 also describes the variables conducted to represent the association between users and attributes.

Table 2.3 Examples of demographic attributes for a user-demographic matrix construction adapted from Vozalis and Margaritis (2004).

No. of Attribute	**Attribute Contents**	**Description**
1	age ≤ 18	each user relates a single age grouping
2	18 < age ≤ 29	the correlating cell assigns value 1 (true)
3	29 < age ≤ 49	the rest of the attributes remains 0 (false)
4	age > 49	
5	male	the cell matching the gender of user is 1
6	female	the other cell takes value 0
7-20	occupation	a single cell describing the user occupation is 1
		the rest of the attributes remains 0

With the above attributes, the system can take the users to analyze and then create the user-demographic matrix construction as illustrated in Figure 2.8.

		Demographic Attributes								
		1	2	3	4	5	6	7	…	20
Users	1	1	0	0	0	1	0	1	0	0
	2	0	1	0	0	0	1	0	1	0
	3	1	0	0	0	1	0	1	0	0
	⋮	0	0	1	0	1	0	0	0	1
	m	0	0	0	1	0	1	0	1	0

Figure 2.8 A user-demographic matrix.

As exhibited in Figure 2.8, binary values are taken to indicate demographic attributes corresponding to the users where 1 represents the demographic attribute belonged to the user and 0 is otherwise. The systems are able to exploit binary variables in the matrix to assign a demographic vector for each user. Finding similarity among users based on their profile can be achieved by using the demographic vector of each user. Vector-based similarity measure such as Cosine similarity has competency to complete this task as same as implementing it with the CF approach. The calculation of Cosine similarity is defined in Equation 2.19.

When the similarity computation is completed, exploiting the outcomes depends on the utilization of recommendation systems. The DF-based neighbors are capable to be used to provide attractions which the active user has not checked-in or integrated with other approaches of recommendation system for enhancement.

In case of strength, DF provides advantage over CBF and CF that it does not require historical data of users' rating for recommendation (Burke, 2002). Nevertheless, the privacy issue makes gathering high-quality personal data very difficult. The insufficient information and low quality of data may lead the system to

provide unpleasant results. Besides, different stereotypes of the user's interests may cause the accuracy and performance of DF is lower than other approaches (Azak, 2010, pp. 12-13). However, DF technique is commonly applied in a hybrid approach to support the other recommendation techniques (Montaner et al., 2003). In this study, the capability of DF approach is adopted to select the users who have similar SNS profiles with the active user. The attention of DF implementation is to reduce the number of users in the mobile engine and enhance performance of the mobile engine in the aspect of response time.

4) Utility-based Approach

Utility-based recommendation systems provide recommendations relied on the calculation of the utility of each item for a user. The techniques of utility-based recommendation adopt features of items as background data, elicit utility functions over items from the user to describe user's preferences, and apply the function to consider the rank of items for the user (Burke, 2002).

The advantage of this approach is that it does not require historical data of users' preferences. Thus, new users, new items, and sparsity do not have any problems with performance of the utility-based recommendation systems (Burke, 2002). Nevertheless, the inevitable problem of the approach is a utility function for each user which should be created. The user is required to construct a complete preference function and weight the importance of each feature that describes an item of interests such as price, quality and delivery date. It could be a burden for the user when he/she must build the function of items that have complex features such as news articles and movies. Hence, considering the way to produce accurate

recommendation with little user effort is necessary when designing the systems (Burke, 2002).

5) Knowledge-based Approach

Knowledge-based recommendation systems try to recommend items based on inference needs and preferences of a user. It exploits knowledge about the user and items in order to reason about what items are appropriate with the user's preferences. Therefore, the systems need to manipulate knowledge including three types as follows (Burke, 2002).

- **Catalog knowledge:** Knowledge about items which can be recommended and their features.
- **Functional knowledge:** Knowledge about how to match between the user's needs and the items that could satisfy those needs.
- **User knowledge:** Knowledge about the users' needs which are required to discover corresponding items.

To reason about what items fit the user needs, the systems need to ask the user requirement of required items. And then it uses answers from the user to exploit knowledge based on the items' domain. Hence, the systems need to have the item domain knowledge for applying with inferring and reasoning. The examples of knowledge-based recommendation systems for tourism were presented by Burke (2002) named Entrée. The system was able to suggest restaurants in a new city that a tourist will visit places where are similar to restaurants the tourist knows and likes in his/her living town. The recommendations adopt knowledge of cuisines to infer similarity between the restaurants. Another system is SPETA (García-Crespo et al.,

2009) adopting knowledge-based approach in order to match between user's preferences and tourism services. In SPETA, ontology was used as knowledge base to store the knowledge of users and tourism services. In order to make inference, SPETA had the reasoner module to do this task.

There are several advantages of knowledge-based approach. For instance, the recommendations systems do not rely on the user ratings of items like CBF and CF. Thus it does not suffer from the cold-start problem. Besides, the system can revise their suggestions rapidly when a user's interests change due to independent of historical user's preferences. Nevertheless, the main drawback of this approach is the need for knowledge acquisition. For example, the user's knowledge acquisition is a very difficult process and a knowledge engineer is required to build the knowledge base (Burke, 2002).

6) Hybrid Approaches

The shortcomings of CBF and CF caused many attempts (Burke, 2002; Pazzani, 1999) tried to overcome. They tried by integrating the two approaches to increase accuracy of recommendation. Pazzani (1999) introduced a framework of recommendation system adopting three approaches, which are demographic filtering (DF), CBF and CF. CBF in case of Pazzani (1999) was used to build user profiles. It also had a role in finding the similarity of between users by determining features of data and rating of user. In this way, Pazzani (1999) attempted to solve the low quality of recommendation from CF approach. Hence, this approach was able to distinguish the similarity between two users although they did not have the co-rated items. However, framework of Pazzani (1999) had some drawbacks occurred with CBF

approach. Such drawback of CBF made it to be able to execute only the comparison with the items that have common features.

In the past, many recommendation systems tried to identify interesting information by employing a sentence matching technique. This technique had limitation affected to decrease quality of recommendation. To overcome this problem, Schiaffino and Amandi (2009) introduced an expert software agent named Traveller. The agent is purposed to support users in tourism domain. Hybrid approaches including CF, CBF and DF are incorporated in this agent to advise weekend trips. Such approach also applied with the WebGuide (Fink and Kobsa, 2002). Besides, there were other approaches adopted to suggest information such as a knowledge-based approach. The knowledge-based approach is capable to infer demands and interests of users. Burke (2002) developed a knowledge-based recommendation system in order to introduce restaurants in unvisited town to tourists. The advice of system was relied on restaurants where the user had known in their hometown. Hence, there was the knowledge base of cuisines adopted to infer the similarity of restaurants. Furthermore, some endeavor aimed to solve the shortcoming of CF. Huang and Bian (2009) and Shih, Yen, Lin and Shih (2011) integrated CBF and CF to recommend attractions. CBF in case of Huang and Bian (2009) was applied with travel behavior of users in the past. This differed from the conventional CBF, which employed description of attractions or activities to perform recommendations. Another approach was taken to deal with behavior of other travelers who shared similar interests to the active user. Shih et al. (2011) took the recommending results obtained from both CBF and CF to consider by implementing Bayesian probability to indicate which one was the most suitable for the user.

This study aims to use the hybrid approach at least two approaches in the proposed mobile engine. These approaches consist of DF and CF. DF could facilitate the mobile engine to identify similar users based on their demographic profile. In case of CF, it may exploit the relationships (Friends) between SNSs members to discover new interests from their friends who have similar preferences, particularly in a tourism domain.

2.2.3 User Modeling

To advise information associated with individual interests of users, the systems are required to execute inference about the user preference. Therefore, it is necessary to reserve historical information of users either behaviors or experiences to establish user modeling (Schiaffino and Amandi, 2009; García-Crespo et al., 2009). Schafer et al. (2000) indicated that recommendation systems tend to present information based on profile or background of user. Hence, every system needs to create and maintain the user model (Montaner et al., 2003).

Rich (1983) explained the characteristics of user modeling distinguished in three dimensions as follows.

1) The dimension about group or individual user modeling

In the first dimension, the system stores a single or multi user models. These models have a role in classifying types of users. When the commercial patterns are more complex that means the kinds of users are categorized with various properties. Skills, demands and knowledge levels are examples of those properties (Ardissono, Felfernig, Friedrich, Jannach, Schafer, and Zanker, 2001). The category of users is called Stereotype or Canonical. Stereotype is a one method adopted to

create user model. This approach relied on assumption that the requirements of an individual may be similar to previous other users. Thus, stereotype-based system can provide personalized information to the user from the beginning. Nevertheless, having similar characteristics can be possible to happen among the users. But in fact, these users could be diverse in other ways. Besides, habit of users may change over time. In contrast, the stereotype-based system is flexible and suitable for dealing individual requirement because it takes a user observing method to build the user model. However, the observing method takes a long time to notice each of users before making the model.

2) The method of user information acquisition

The second dimension is a user information acquisition method to form user models. The user information acquisition method is based on the view of Rich (1983), which is divided into two classes. The first one is explicit user models. Recommendation systems depended on the first approach which needs users to fill in their profiles. After that, the systems exploit the stored data to present information correlating with users' profiles. Nonetheless, the explicit acquisition can bother the users when they are required to fill out too many questions. Furthermore, the users sometime deliver the incorrect information to the system therefore it causes the system to perform recommendation with unsatisfying results. The second one is implicit user models. This class is more reliable and friendlier to the users than the first one. However, creating assumption of preferences for each user could not be perfectly accurate. Besides, the system does not have adequate time to observe all users to synthesize the assumption of users with high precision.

Even though, the two acquisitions have different advantages and disadvantages, there is a tourism recommendation system named SPETA taking both explicit and implicit approaches to facilitate each other (García-Crespo et al., 2009). Besides the explicit and implicit acquisition, Kabassi (2010) mentioned the exploitation of personalized information on SNSs. The information, particularly user profiles and user behaviors occurred within SNSs, could be extracted to apply with the recommendation system.

3) The time period of the user models

The last dimension concerns time period of user models between short-term and long-term. Because historical interaction of active user and similar users is crucial information for DF, CF and CBF, most of recommendation systems preserve the user models in long-term. Besides, the long-term user models are commonly adopted in tourism recommendation systems in order to provide suggestion with more effective.

The perspective of Rich (1983) reflects three dimensions to acquire user models. According to the first dimension, this research intends to create user models from a group of users due to the nature of SNSs as networking of people. Both explicit and implicit user models from the second dimension can be used to apply in this research. Due to the unique characteristics of SNSs especially user profile, this part demonstrates user demographic data such as gender, birthdate, language or geographical location. Therefore, it can exploit these data to make explicit user models. In case of implicit user modeling, social functions on SNSs such as Like and Check-in are capable to imply the interesting attractions based on these functions. The final dimension is time period of user modeling. This study aims to

maintain user models in long-terms because these models could enable the mobile engine to predict requirements of users in the future.

2.2.4 The System Evaluation

Evaluation is an important part of system development in order to approve the correctness. Chin (2001) stated that system assessment was necessary to determine which users are helped or obstructed by their historical interaction in the user modeling system. He also mentioned that good empirical assessment depended on the appropriate experimental design and manipulation, particularly unique factors. These factors were able to be isolated from other confusing factors. Nonetheless, Chin (2001) remarked that the empirical experiment did not frequently occur in literature of user modeling. Many researchers also mentioned that evaluation was a significant part. It was conducted to prove the better feedbacks of the systems including user modeling when they were required to compare with non-user modeling systems (Chin, 2001; Micarelli and Sciarrone, 2004; Cheng and Vassileva, 2006).

Tourism recommendation systems can be evaluated by human and machine. Gulliver's Genie (Hristova, O'Hare, and Lowen, 2003; O'Grady and O'Hare, 2004) and m-ToGuide (Kamar, 2003) conducted several evaluation by human as users such as satisfaction of users, rightness of recommendation as well as possibility of development. On the other hand, Coyle and Cunningham (2003) estimated their system named PTA by using simulation of interaction. The simulation was established from users' historical interaction. However, there was an evaluation of system based on combination of two approaches introduced by Yap, Tan, and

Pang (2005). In this case, they used both real human and virtual human as computer-generated users to operate assessment.

Considering the performance of personalization, it supports an engine to raise in assessment of Personalization Travel Support (PTS) system. To evaluate effectiveness of PTS, Srivihok and Sukonmanee (2005) applied measurements from information retrieval (IR) such as precision, recall and harmonic mean (F-Measure) function. Precision is the ratio of interested trips over the total number of suggested trips. To compute the precision, the number of trips clicked by users from the suggested trip is divided by the number of suggested trips. Meanwhile, recall is the ratio of interested trips over the total number of clicked trips. Calculating recall is dividing the number of trips clicked by users from the suggested trip by the number of clicked trips in user's transaction. The last one is F-Measure. It is used to express the performance of integrating precision and recall through harmonic mean function. F-Measure can be computed by taking the result from multiplying precision and recall to divide by the sum of precision and recall. F-Measure is assumed that if it has a high value which means precision and recall are high as well.

The calculation of *Precision*, *Recall* and *F-Measure* values can be calculated by the equations presented in Miao, Duan, Zhang, and Jiao (2009). There are four data of recommendation system used in the calculation. As demonstrated in Table 2.4, these data consist of True Positive (*TP*), False Negative (*FN*), False Positive (*FP*) as well as True Negative (*TN*). *TP* represents the number of relevant data retrieved from a system. *FN* is the number of relevant data that a system does not regain (missing results). The *FP* is the number of irrelevant data recovered from a system (unexpected results). The *TN* expressed the number of irrelevant data which

does not retrieve from a system. The calculation of precision, recall and F-Measure is shown in the Equations 2.23, 2.24 and 2.25, respectively.

Table 2.4 Confusion matrix represented evaluating effectiveness and efficiency of recommendation system based on information retrieval measurements.

Predicted Category	**Actual Category**	
	Relevant data	**Irrelevant data**
Retrieved Data	*TP*	*FP*
Not Retrieve Data	*FN*	*TN*

$$Precision = \frac{TP}{TP + FP} \tag{2.23}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.24}$$

$$F - measure = \frac{(2 \times Precision \times Recall)}{(Precision + Recall)} \tag{2.25}$$

There are three evaluations of the mobile engine comprising performance of category prediction, correctness of recommendation and response time. These evaluations are relied on real human as users and experts. The first one adopts recall to measure correctness of predicted attractions using LSA and ML techniques. The second one exploits recall to indicate performance of recommendation. Recall is used to assess system's ability of relevant document

recovery. This study adopts the Equation of recall from Miao et al. (2009) to operate the second appraisement. The recall is calculated by the Equation 2.23 as displayed above. The response time is represented the processing time when the mobile engine manipulates its processing. Besides, this evaluation would reflect the wall-clock time that the users will obtain the final feedbacks. Further details of three evaluations will be explained in Chapter 3.

2.3 Social Networking Sites

The popularity of SNSs such as Facebook, Twitter, Google+, LinkedIn, MySpace, and Path has attracted million people to participate and share their interests, experiences, preferences as well as lifestyles. SNSs enable users to generate their contents and create relationship with others freely. This makes these services plenty of user generated contents. Due to the richness of information, SNSs become one of enormous online personal data's resources. In this study, SNSs are very interesting resources to extract and identify the interests of users in tourism domain. These extracted data have benefits to build and maintain a user model in the recommendation systems for tourism. Hence, this section provides definition and characteristics of SNS, social network connect services, and Facebook application architecture, respectively.

2.3.1 Social Networking Site Definition

Boyd and Ellison (2007) defined social network sites as web-based systems which allow individuals to 1) build public or semi-public profiles within a bounded system, 2) communicate with a list of other users who shares a connection,

and 3) view and visit their connections as well as others' connections in the system. In each SNS, those connections are defined and called in various ways.

In research of Boyd and Ellison, they preferred to use the term of social network site rather than adopting the term of social networking site. They said that the term of social networking site was building a relationship with strangers on SNS which no users do this. Commonly, SNS users tend to create the relationship with people they have already known. Nevertheless, Beer (2008) argued that the term of social networking site is more appropriate than another one because the term of social network site would be too broad to reach the similar meaning of Web 2.0. To support perspective of Beer (2008), there are definitions of social networking site from Dictionary.com, Oxford Dictionaries and Cambridge Dictionaries Online as follows.

Social networking site is a website that allows subscribers to interact, typically by requesting that others add them to their visible list of contacts, by forming or joining sub-groups based around shared interests, or by publishing contents so that a specified group of subscribers can access it (Dictionary.com, www, 2012).

Social networking is the use of dedicated websites and applications to communicate informally with other users, or to find people with similar interests to oneself (Oxford Dictionaries, www, 2012).

Social networking site is a website that is designed to help people communicate and share information or photographs with a group (Cambridge Dictionaries Online, www, 2012).

Even though, the definitions of SNS from those dictionaries did not mention to the building association with strangers, the similar interests among the

users are a noticed issue. Currently, SNS users do not only construct connection with the recognized people, but they also join the groups of users who share similar interests to the users. In those groups, the users may not know each other before. This study selects the term of social networking site or social networking service. However, the explanation of social networking sites' characteristics in the next subsection is based on the perspective of Boyd and Ellison (2007).

2.3.2 Characteristics of Social Networking Sites

SNS ordinary is driven by its users and it also has different natures from earlier online communities and websites. Those earlier services are organized on interests and topics. According to Boyd and Ellison (2007), there are three unique characteristics of SNS which make it different from online community as follows.

1) Profile

Although current SNSs have many dissimilar features, the core feature is profiles. The SNS users have profiles to contact with other users in the system. Profile is a unique page which the SNS users use it to express themselves (Sunden, 2003). Basically, the new user starts with the system by developing his/her profile. The user profile consists of demographic information such as age, location, workplace as well as personal interests. In some SNS, there are some series of question for the first time usage. Many SNSs encourage its user to upload a profile photo. Besides, some of the SNSs allow users to use multimedia for editing their profile. Some SNSs such as Facebook permit its users to add applications in order to adjust their profiles.

2) Friends

After a user becomes a member of SNS, the user will be stimulated to create relationships with other members. The term of relationship on each SNS is different such as “Friends”, “Contacts”, “Fans”, and “Followers”. Establishing the association between users mostly needs to respond each other. Nonetheless, some SNSs do not need these responses. Hence, those SNSs prefer to use “Fans” or “Followers” as the term of one-sided relationship. The term "Friends" can be misleading, because the connection’s creation on SNSs is not necessary to percept on both sides as in real life. Hence, this makes the users be able to contact each other within SNSs diversely.

When the relationship had been established, the user can view his/her Friends’ profiles. Furthermore, the user is able to visit and invite Friends of Friends to join their social. Typically, every field of user profile can be searchable because SNS enables the user to seek others who share common interests and backgrounds. Viewing the user profile depends on the system design and privacy setting of the user.

3) Comments and Private Messaging

Essentially, SNSs have a tool that lets users to leave their messages on their Friends profiles. These messages are typically called Comments. Besides, SNSs provide a tool which grants users to send and receive private messages as well as e-mail service on Webmail. Both tools are very popular on major SNSs.

Besides the three specific characteristics of SNSs, there are other characteristics which each SNS raises them as remarkable features to be different from its competitors. Examples of those features include photos sharing or videos

sharing, blogging and instant messaging. Some SNSs such as Facebook, MySpace and Twitter are extensively developed on mobile platform. Furthermore, some SNSs grant its users to install additional applications.

In summary, SNSs commonly have three unique features including: 1) Profile page is used to describe an identity of user. The profile page comprises user's demographic information, user's interests or preferences and a list of Friends. 2) Term "Friends" is used to call relationship between the user and others on SNS. Each of SNSs has different terms to call the establishment of relationship. 3) SNS users are able to leave their comments on their Friends profile. They can also send and receive private messages.

The unique characteristics of SNS reflect its benefits to enhance the recommendation system in this research, especially profiles and Friends. The profile enables the recommendation system to harvest up-to-date information of users such as demographic data, interests and lifestyles. Besides, the demographic data is able to be adopted in order to identify the users who share similar profiles. In case of social relationship, the recommendation system can exploit the relationship to find similar users who share common interests. In addition to profiles and Friends, the geographic information such as attractions or POIs can be found on SNSs. Due to the nature of SNSs as the system driven by people, there are many generated contents from the users and one of them is those attractions.

2.3.3 Social Network Connect Services

According to Ko, Cheek, and Shehab (2010), the SNS user data basically includes three main kinds of information. The first one is Identity data used

to explain who I am in SNS. It consists of user identity, user profile as well as privacy policy. Another is Social-graph data. This data has a role in representing who I know in SNSs, especially a list of user's friends with description such as family, coworker and colleague. The last one is Content data adopted to indicate what I have in SNSs. These data are composed of user messages, comments, photos, videos, and contents generated by the SNS users.

Commonly, many SNSs allow third-party sites to exploit data on SNSs with their own sites. Therefore, the security and reliability of SNSs is necessary. As shown in Figure 2.9, there are four following categories of APIs which grant the third-party sites to exchange the information with SNS.

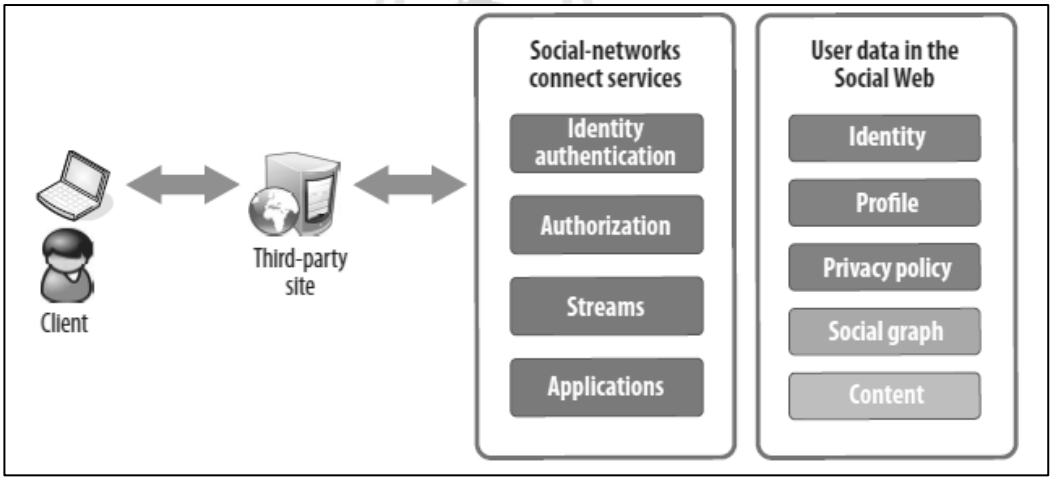


Figure 2.9 The regular social-networks connect a service framework (Ko et al., 2010).

1) Identity authentication

To prove identity of users, the users need to authenticate by using their existing accounts to admit the SNSs.

2) Authorization

This API plays an important role in manipulating user data access on SNSs. Hence, predefined accessing right should be achieved before accessing. The authorization also allows third-party sites to establish new contents and extract existent contents from SNS users.

3) Streams

The stream API permits third-party sites to publish users' activity stream, particularly posting any contents in SNS.

4) Applications

The last APIs allow third-party sites to reach plenty of SNS features and exploit them to develop in the form of applications. These applications tend to expand both ability of SNS and a number of users.

Moreover, Ko et al. (2010) proposed the Facebook Platform services relied on the framework of social-network connect services as illustrated in Figure 2.10. The Facebook Platform has become available to public since 2008. The famous Facebook API such as the Open Graph was launched in April 2010. Such API now is widely used by many third-party sites in order to incorporate between those sites and Facebook. This facilitates third-party sites to exploit rich information and extend Facebook capability with developing further applications.

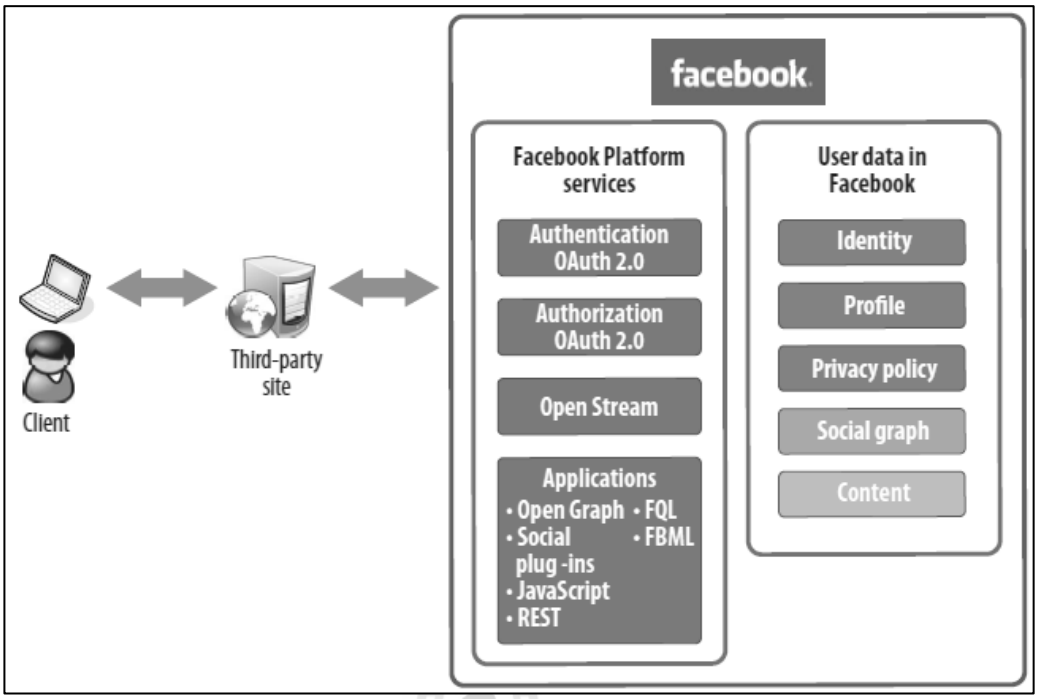


Figure 2.10 Facebook platform services (Ko et al., 2010).

Facebook allows the users to manage their data via the third-party sites such as identity, profile, privacy policy, social graph, and content. In Figure 2.10, Facebook platform offers many APIs for supporting third-party sites to cooperate with it. For example, OAuth 2.0 protocol is provided for authentication and authorization by using Facebook account. The protocol allows Facebook members to use their Facebook account for authenticating on the third-party sites supported this protocol. For instance, Wongnai.com, a restaurant recommendation website which users can share and comment restaurants, accepts using Facebook account to register for new members as shown in Figure 2.11. In case of authorization, OAuth 2.0 lets third-party sites obtain authorization tokens from Facebook after user authentication was finished. When the third-party sites have obtained the tokens, they can request

further permissions relied on the specific requirements of the applications such as accessing user profiles and publishing contents on the user wall as shown in Figure 2.11.

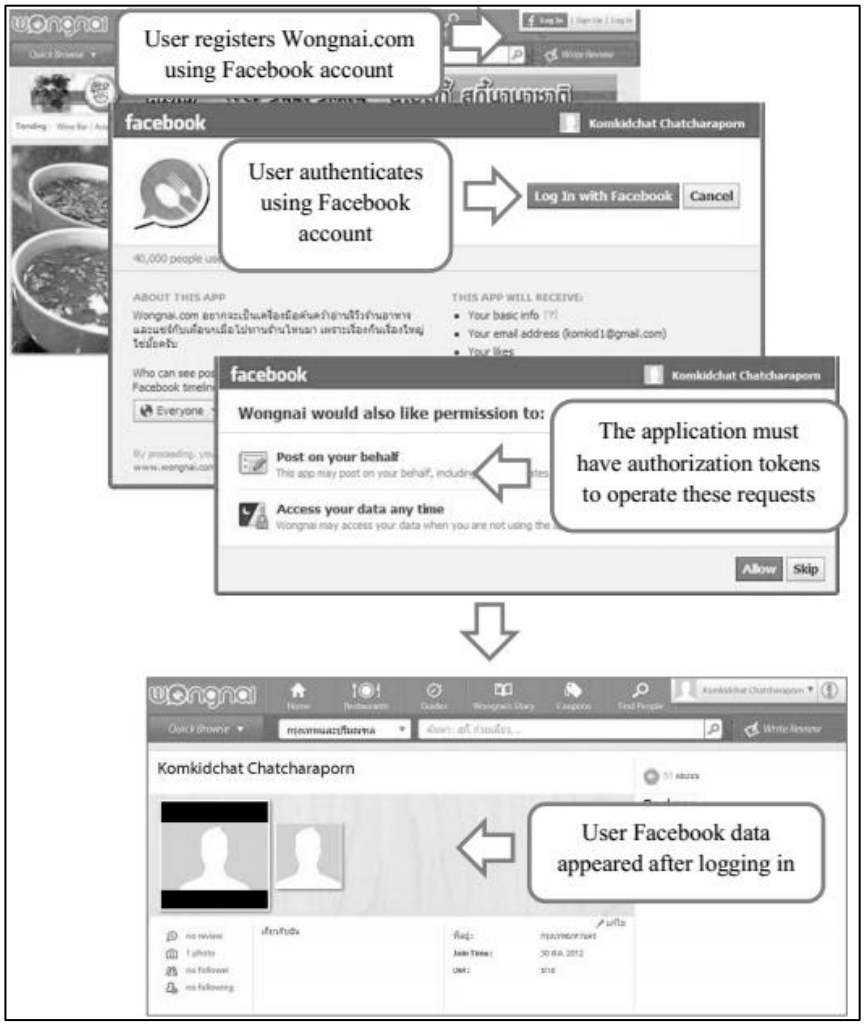


Figure 2.11 An example of Wongnai.com authentication based on Facebook Platform. A new member is able to use their Facebook account to authenticate (steps 1, 2 and 3). After logging in, the user can edit his/her information in Wongnai.com and share restaurants from this site to their wall.

Besides, Facebook has an Open Stream API in order to allow third-party applications to read and write users' activity streams such as posting comments on photos of their friends. The API also supports multiple-stream publishing and the Atom feed standard. Releasing Open Stream API lets the Facebook members read and write their activity streams via any third-party sites supporting the API.

Facebook offers various APIs to support third-party sites implementing core features of Facebook, especially read and write data. Such Open Graph API is extensively implemented with many Facebook applications to access the information in Facebook through URLs. Social plug-ins enable the traditional website to extend with Social functions of Facebook such as Log in with Facebook account and the Like button. In addition, Facebook has JavaScript including classes and methods in order to adopt in the third-party application. The conventional API of Facebook such as Representational State Transfer (REST) now is replaced by Open Graph API. Furthermore, Facebook provides powerful features named Facebook Query Language (FQL). FQL enables developers to query data on Facebook as same as using SQL. The queried feedbacks can be either in eXtensible Markup Language (XML) or JavaScript Object Notation (JSON) formats. The last feature is Facebook Markup Language (FBML) which is used to render webpages on Facebook platform like HTML.

In this subsection, social network connect services are presented to show the ways of accessing user data in SNS. The Facebook Platform based on social network connection services provides the unique features to support third-party applications or websites in order to extend with the Social web. In this study, Facebook Platform is implemented to facilitate a Facebook application in the mobile

engine employed to explore user interests in the tourism domain. Therefore, applications as a part of Facebook platform will be explained in further details in the next subsection.

2.3.4 Facebook Application Architecture

This subsection presents Facebook Application Architecture to understand how Facebook applications facilitate the mobile engine in this study.

Basically, web applications are based on Client-Server architecture. As exhibited in Figure 2.12, client starts to work by forwarding HTTP request to a web server. After that, the server processes data based on the request and sends feedbacks as HTML back to display in the client. However, Facebook application architecture is different because it has Facebook server intermediated between client and server as demonstrated in Figure 2.13. In Facebook application architecture, the Facebook server acts as a middleman, which has a role in receiving requests from client-side. And then, the Facebook server will transfer the request to a web server as third-party site. When the web server is operating, it is able to connect with the Facebook server via available APIs (e.g. Open Graph, FQL) if it requires additional data. After the web server finishes its process, the output will be transmitted to the Facebook server in FBML format. Finally, the Facebook server converts the outputs from the FBML to HTML format in order to display on the client.

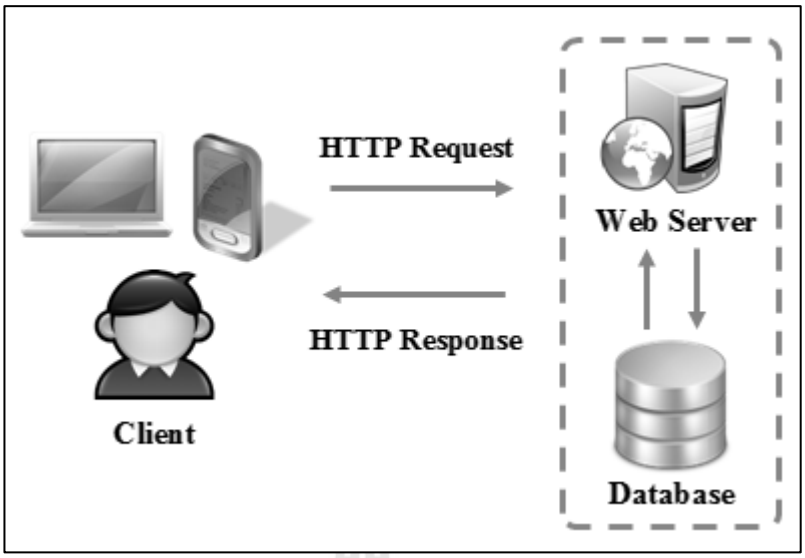


Figure 2.12 Regular website architecture adapted from Graham (2008).

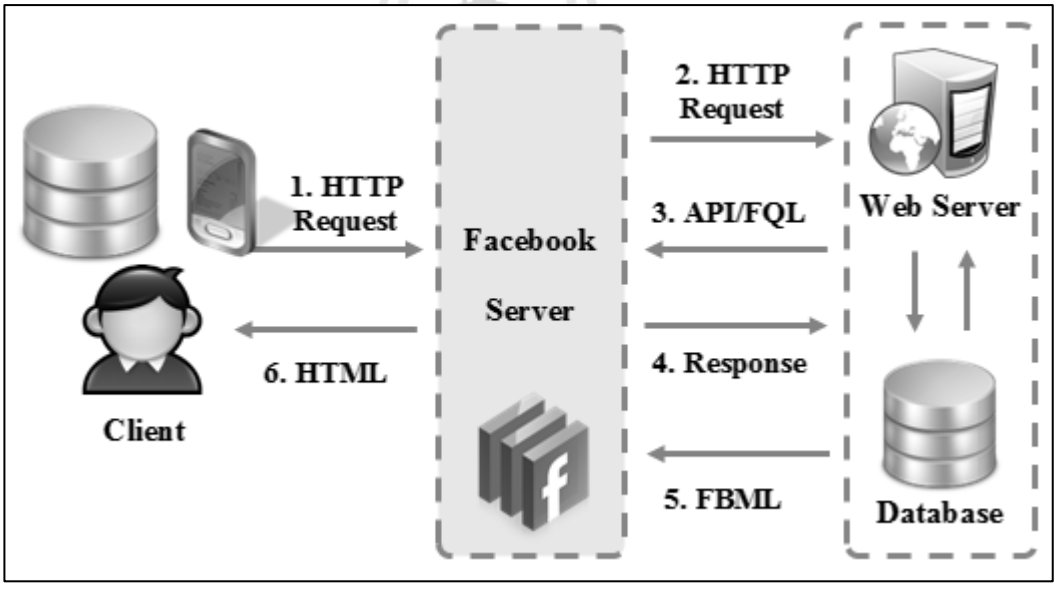


Figure 2.13 The architecture of Facebook application adapted from Graham (2008).

According to Ko et al. (2010), the available services in Facebook platform, especially the applications include six APIs as displayed in Figure 2.10.

These components enable developers to integrate their application with Facebook platform conveniently. The following explanations are the six components in details.

1) Open Graph is primary API for the third-party applications. The popularity of this API comes from its simple usage. It allows the third-party applications to read and write contents on Facebook (e.g., Photos, Music, Newsfeed, Friends as well as Check-in histories) through URLs. For instance, using the website “https://graph.facebook.com/komkid1” will provide the information of user named “komkid1” back in JavaScript Object Notation (JSON) format. The JSON format is extensively used as a main format of feedback in many SNSs because it is easy to implement with any programming languages. An example of JSON as a set of feedback by adopting Open Graph API is displayed below:

https://graph.facebook.com/komkid1

{
"id": " 713536936",
"name": " Komkid Chatcharaporn",
"first_name": " Komkid",
"last_name": " Chatcharaporn",
"link": "http://www.facebook.com/ komkid1",
"username": " komkid1",
"gender": "male",
"locale": "en_US"
}

In addition to users' information, there is a set of objects which developers can reach via Open Graph API. These objects include:

- Pages,
- Events,
- Groups,
- Applications,
- Status messages,
- Photos,
- Photo albums,
- Profile pictures
- Videos,
- Notes,
- Check-ins.

2) Social plug-ins allow the conventional websites to attach some social functions in the form of minimal HTML codes to connect with Facebook. Like button is an obvious example of social plug-ins appeared on pages to let the users share those pages and send them back as posted contents on their Facebook profile. Another example is Login button. This button permits the users to take their Facebook account for accessing any websites which install these social plug-ins. Besides, Facebook provides various social plug-ins such as Newsfeed, Comments and Fanpages.

3) JavaScript as a software development kit, which includes classes and methods that third-party sites can implement to support their Facebook applications. Originally Facebook had its own JavaScript named FBJS. However, when the Open

Graph API was released, Facebook announced that it will not support FBJS and persuade developers to use the Open Graph API instead.

4) Representational State Transfer (REST) is traditional API of Facebook. In the past, REST was a primary API employed to enter core services of Facebook such as users' profile, a list of Friend, photos, and videos. Furthermore, it was manipulated Facebook functions like logging-in, redirecting and updating-views. Nevertheless, REST API helped Facebook to become the first SNS which allowed developers to develop applications associated on Facebook resources. Hence, there were many applications on Facebook based on REST API and now they were shifted to implement with Open Graph API. As same as the FBJS, the emergence of Open Graph API eventually reduces the significant of REST (Ko et al., 2010).

5) Facebook Markup Language (FBML) as HTML of Facebook is defined for rendering a web page on Facebook Canvas (Graham, 2008).

6) Facebook Query Language (FQL) as SQL for Facebook Platform that enables developers to query Facebook user data in order to use them with their applications comfortably (Graham, 2008).

Facebook application is an important component in this research due to the capability of cooperating between Facebook server and the third-party site. The available APIs in the application, especially Open Graph API and FQL, facilitate Facebook application in the mobile engine to retrieve the users' interest data in tourism domain. Nonetheless, in order to retrieve data, the Facebook application must have authorization tokens to perform the request every time as mentioned in the previous subsection.

2.4 Location-based Social Networking Services

The emergence of location acquisition technology on mobile device such as GPS and WIFI enables users to attach a dimension of location with existing SNSs. For instance, users are able to upload location-tagged photo to SNS such as Flickr, Twitter and Facebook. These locations in SNSs can be represented in the form of coordinates (e.g. latitude and longitude) or icons (e.g. restaurants, accommodations, and coffee shops). According to Zheng and Zhou (2011), the location's dimension shows the possibility to bring users in the virtual world back to the real world such as inviting people in SNSs to a location for enjoying activities together. Furthermore, the dimension facilitates users in existing SNSs to extend their network with the new interdependency obtained from their locations. The interdependency includes friendship, mutual interests and participated knowledge. Hence, the location can be adopted to infer explicit knowledge and individual interests of users by learning their locations.

The location-embedded and location-driven social networks are called location-based social network (LBSN). In this section, the definition of LBSN is described followed by three categories of location-based social networking services (LBSNSs). Lastly, the explanation of a famous social function of LBSNSs named check-in is presented.

2.4.1 Location-Based Social Network Definition

The formal definition of location-based social network (LBSN) was coined by Zheng and Zhou (2011). They stated that a LBSN did not merely integrate a location into an existing SNS in order to let the people in the network to share their

location attached information, but it also includes the novel social structure which is created by individuals and associated by the interdependency of users. The interdependency is acquired from the physical locations of users where they visit and their location-tagged media content, such as texts, photos as well as video. Those physical locations comprise the instant location of an individual including timestamp and history of location. Normally, the instant location was accumulated by the individual in a certain period. In case of the interdependency, it does not only consist of co-occurrence between two persons at the same physical location or share similar location histories, but it also includes the knowledge of people such as mutual interests, behavior and activities. The knowledge can be implied from individual's location in history and location-tagged data.

2.4.2 Categories of Location-Based Social Networking Services

According to Zheng and Zhou (2011), there are three distinguished categories of LBSNSs provided in the current applications. These categories include geo-tagged-media-based service, point-location-driven service as well as trajectory-centric service.

1) Geo-tagged-media-based service

The first one enables users to attach digital media to a location such as text, images and videos created in the real world. The attachment can be achieved immediately after the media is established or when the users have returned their homes and do it later. In this case, the users are able to view their content at the location attached with digital media by using a digital map or an augmented reality (AR) application on mobile devices. Besides, the users are capable to comment on

those media and extend their social networks adopting interdependency obtained from the geo-tagged content. For instance, the same photo is captured at the same location. The example sites providing geo-tagged service are Panoramio, Flickr and Geo-twitter. Nonetheless, dimension of location is merged into SNSs. The services merely focus on the digital media content. Therefore, location is merely a feature in order to fulfill the media content which is the main interdependency between the users.

2) Point-location-driven service

Foursquare, Google Latitude and Facebook Places are location-based services which stimulate the users to share their current locations like sightseeing, restaurants or historical sites. In case of Foursquare, game elements such as points, badges, and reward are adopted for stimulation when the users are "checking in" at the venues. The person who has the most number of "check-ins" at a venue is called "Mayor". The current location sharing in real-time allows the active user to explore his/her friends from LBSNSs in physical world and join some activities together such as asking people to watch movie or have dinner. Another function of Foursquare such as "tips" lets the users to suggest or recommend some tips to venues for the others to read it. In the second type of LBSNSs, a venue (point-location) is the main component to consider the interdependency connecting users, while the user-generated content such as tips or badges is a feature of venue.

3) Trajectory-centric service

The third category like Bikely, SportsDo and Microsoft GeoLife lets users to concentrate on both point locations passed by a trajectory and the detail of itinerary associating with these point locations. The users do not only reach basic

information in these services (e.g. distance, duration, and velocity of trajectory), but they also demonstrate their experiences obtained from itinerary via tags, tips and photos. In summary, besides “where and when” information, the third services deliver “how and what” information. Hence, the other users are able to approach the shared experience of a user by exploring the trajectory on a digital map and tracking the trajectory in the physical world with GPS function on mobile phone.

Zheng and Zhou (2011) illustrated the comparison of these LBSNSs categories as shown in Table 2.5.

Table 2.5 Comparison of the main categories of LBSNSs (Zheng and Zhou, 2011).

LBSN Services	Focus	Real-time	Information
Geo-tagged-media-based	Media	Normal	Poor
Point-location-driven	Point location	Instant	Normal
Trajectory-centric	Trajectory	Relatively Slow	Rich

In this research, the point-location-driven LBSNS especially Facebook Places acts as a main data source for providing “checked-in” information of SNS users. The information may lead to many benefits in order to improve performance of tourism recommendation system. However, due to user-generated content, Facebook have plenty of incorrect location information. Hence, LBSNS such as Foursquare plays an important role in revising the categories of retrieved locations from Facebook. This study does not take “checking-in” information from Foursquare into consideration because some users of the mobile engine may not have an account on Foursquare.

2.4.3 Check-in

The prior subsection introduced types of LBSNs and the term of check-in was mentioned in the point-location-driven services. The users can check-in as announcing their current location to their friends through the social networks (Vasconcelos, Ricci, Almeida, Benevenuto, and Almeida, 2012). People tend to perform checking-in with current locations where they are visiting. In LBSNs, these locations called venues represent the places in the real world such as coffee shops, restaurants, museums, or even commercial brands and business companies. The famous LBSNs such as Foursquare and Brightkite convert check-ins into points like playing game. These points allow the users to receive badges or become Mayor.

According to Richmond (2010), users are able to check in a specific location by forwarding text messages or adopting mobile applications on their mobile phone. Such applications exploit GPS function of mobile phone to detect the user's current location. Besides, many of them offer a "Places" button or tab which users can press and view a list of nearby locations where are available for checking in. If a place does not appear in the list, the user can create the place via the application instantly. After checking in, users can comment some tips with those checked in venues and share the information to their friends in other SNSs such as Twitter and Facebook.

This study exploits a check-in function of LBSNs to discover the interesting locations of users where they have visited. The historical checking in could be applied to both individual interests and the mutual interests of individual and his/her friends in the tourism domain. Hence, the performance of checking in may lead to the possibility to enhance efficiency of tourism recommendation system.

2.5 Related Work

The rapid growing of the Internet usage causes tourists to suffer from the information overload problem inevitably. The development of recommendation system is one solution adopted to overcome the information overload problem by providing personalized information (Shih et al., 2011; Huang and Bian, 2009). Furthermore, the recommendation system assists tourists to filter unnecessary information and support decision of travel planning. According to Huang and Bian (2009), there are many stages of travel planning such as selecting destinations, choosing tourist attractions, selecting accommodations, considering routes, etc.

In order to deliver personalized recommendations, the system needs to infer users' profiles or preferences and match them with available contents. Basically, there are two methods implemented in the system for extracting user data (Gavalas and Kenteris, 2011; Kabassi, 2010; García-Crespo et al. 2009; Schiaffino and Amandi, 2009). The first one demands explicit feedbacks of the users to consider their interests such as rating given to attractions, ordering interest travelling places from the most to the least, and criticizing preferred contents. The other one is an implicit method which tends to exploit user interaction with the system in order to make inferences such as visiting background, recording visited contents, monitoring users' behaviors of selecting contents. Most existing tourism recommendation systems tend to apply both methods in order to acquire the user preferences (García-Crespo et al., 2009; Schiaffino and Amandi, 2009; Huang and Bian, 2009; Gavalas and Kenteris, 2011; Hsu, Lin, and Ho, 2012).

In recent years, the popularity of SNSs has become the interests of scholars in the field of tourism recommendation system. Therefore, many scholars exploited a

large amount of information in SNSs to identify travelers' preferences manipulated in systems, especially the interests of their social network and their behaviors. SPETA (García-Crespo et al. 2009) implemented a SNS named OpenSocial by taking user profiles and contact lists of active users into consideration of recommendations. Besides, with the rapid advancement of mobile devices and ubiquitous Internet access, location-based SNSs (LBSNSs) have emerged in recent years. These LBSNSs such as Foursquare, Gowalla, Whrrl, Brightkite, and Facebook Places have attracted millions of people to share their social friendships, experiences and tips of POIs via check-ins. The emergence of LBSNSs has stimulated attention of researchers to exploit check-ins of users to achieve POI recommendation. The check-in function of LBSNSs can be used to extract traveling behavior of users and take them to imply users' preferences (Ye et al., 2010; Berjani and Strufe, 2011). Both extracted information from SNSs and LBSNSs are classified into the implicit way because they do not represent the user preferences obviously. Hence, there were numerous preference definitions illustrated by those attempts such as comparing users' SNS profile with traveling services, adopting frequency of checking-in as rating, and using GPS trajectories. In case of LBSNSs according to Zheng and Zhou (2011), they distinguished characteristic of LBSNSs into three types. The difference of LBSNSs categories is explained in subsection 2.4.2.

According to Cheng et al. (2012), currently the study of POI recommendation is divided into two lines. The first line mainly focuses GPS trajectory logs of several hundred monitored users for POI recommendation (Xiao et al., 2010; Zheng, Zhang, Ma, Xie, and Ma, 2011). Basically, GPS trajectory data comprises small number of users but its data is dense Zheng et al. (2011) introduced GeoLife in 2007. GeoLife is

a GPS-log-driven application on Web maps. It also adopted to collect GPS trajectories of users in order to be conducted as datasets for many work of POI suggestion relied on GPS trajectory data. The other line concentrates on LBSNSs data which are very sparse and large-scale (Ye et al., 2010; Ye et al., 2011; Cheng et al., 2012). There were many LBSNSs conducted in research of POI recommendation. Gowalla was a LBSNS website created in 2009 and closed in March 2012 after acquired by Facebook in late 2011. Presently, Gowalla provides public APIs which enable researchers to crawl all users' information consisting of all check-in history with time stamp and detail of locations. Many studies conducted the Gowalla data to test their assumptions in order to enhance competency of POI recommendations (Berjani and Strufe, 2011; Rahimi and Wang, 2011; Cheng et al., 2012; Cheng et al., 2012a; Ying et al., 2012; Yuan et al., 2013; Zhang and Chow, 2013; Picot-Clémente and Bothorel, 2013; Zheng et al., 2013). Foursquare was the second most widely used LBSNS in research of POI suggestion. Some work took other LBSNSs such Whrrl (Ye et al., 2011) and Brightkite (Wang, Terrovitis and Mamoulis, 2013) to perform POI recommendation. Observe that the previous work did not mention to conduct Facebook Places for POI recommendation although Facebook had more users than the others. Facebook released Facebook Places as its LBSNS in 2010 and it was not native LBSNS like Foursquare and Gowalla. However, the strength of Facebook is a plenty of users' profiles and interests which will be investigated in this study along with users' check-in histories obtained from Facebook Places.

According to Bao, Zheng, Wikie and Mokbel (2013), there are three data sources adopted in POI recommendation systems for LBSNSs. The first one is user profiles such as age, gender, interests, preferences, etc. Another one is user geo-

located content including a user's ratings of visited places, geo-tagged contents and check-in histories. Other one is user trajectories comprising sequential locations stored in a user's GPS trajectories. All work except the work of Zheng et al. (2011) manipulated user check-in histories to achieve POI recommendations. Zheng et al. (2011) extracted GPS trajectories of users from GeoLife to operate POI recommendation. Commonly, LBSNSs tend to store check-in behaviors of user and their social relationship. Nevertheless, these services keep a few attributes of use profile such as name, gender and birthdate. This shows the possible way to try the plenty of user profile from SNS such Facebook to consider with the user check-in histories from LBSNSs in order to improve the quality of POI recommendations.

After acquiring user data from above data sources, the system requires to operate recommendation by using information filtering approaches. The widely adopted approaches in existing tourism recommendation systems include DF, CBF, CF, and hybrid approaches. Nonetheless, the most widely adopted approach for POI recommendation based on LBSNSs is CF. Basically, the CF approaches can be divided into two categories (Hsu et al., 2012, Schiaffino and Amandi, 2009) as follows.

Memory-Based CF approaches adopt nearest neighbor algorithms which consider a group of users who shared similar interests with active users. Then they adopt the preferences of neighbors to perform prediction or recommendation for the active users.

Model-Based CF approaches use statistical or machine learning approaches to analyze user's historical records for creating a preference model. The model is exploited to provide predictions. Common approaches used to execute model-based

CF approaches include clustering, association rules, Bayesian networks, or regression analysis.

In addition, model-based CF approaches also include the probabilistic generative model-based method and matrix factorization-based method (MF). The probabilistic-based methods tend to be implemented with geographical influence of check-in that will be explained in the following paragraphs. In case of MF-based CF, it mostly exploits a singular value decomposition (SVD) model to predict ratings of unvisited locations for the active users.

Adopting the above algorithms for achieving CF-based POI recommendation depends on system implementation with the three influence factors: 1) Geographical influence, 2) Social influence, and 3) Temporal influence. Those related work had defined significant of the three influences of user check-in behavior in different ways. However, the core concept of these influences is the same. Geographical influence associates with the distance between the users and locations. The distance affects to the decision of users for checking-in. Social impact is based on the assumption that friends might share a lot of mutual interests which could be correlated with check-in behaviors of users. Temporal influence considers that time affects check-in behaviors of users in the aspect of types of visited location. Most of prior attempts tended to fuse these influences with various CF approaches in order to improve performance of POI recommendation. The following details explain the implementation of POI recommendation with various influences and CF approaches in the previous related work.

A pioneer work of POI recommendation in LBSNS was proposed by Ye et al. (2010). The work investigated social and geospatial impacts to performance of

suggestion. They presented two approaches named a friend-based collaborative filtering (FCF) and Geo-Measured FCF (GM-FCF). The two approaches were performed with LBSNS data fetched from Foursquare. The intention of this work was to overcome computational overhead of memory-based CF. Hence, the first approach was considering solely social friends and the second approach was determining the distance between friends for friend selection. The evaluation results revealed that the proposed approaches were able to reduce computational cost of CF recommendation. However, they also decreased recall because some of dominating users were removed. This work was extended and further studied in the next year. Ye et al. (2011) proposed a fusion framework of USG (User, Social and Geographical influences) including three models in order to improve accuracy of suggestion. The three models were a user-based CF, a social influence model and a geographical influence model. The user-based CF was exploited to compute score of recommending locations for the active users. The social influence model employed friends of the active users rather than took all of users to perform POI recommendations. The geographical influence model adopted a power law distribution to estimate the probability of check-in at given distances from the previous visited locations of the active users. The geographical influence in this work was based on the assumptions that 1) people tend to visit POIs close to their homes or offices; and 2) people may be interested in exploring POIs of a POI that they like even though it is far away from their homes. Foursquare and Whrrl were two data sources gathered to test with a USG framework.

Berjani and Strufe (2011) attempted to adopt a Regularized Matrix Factorization (RMF) technique for POI recommendation. To challenge disadvantage

of memory-based CF, the proposed technique applied regularized Singular Value Decomposition (SVD) to create a model for predicting unvisited locations to the active users. This research extracted check-in histories of users from Gowalla and there was no mention of the three influences associating with check-in behavior of LBSNSs users. Instead of providing the exact location for the active users, Rahimi and Wang (2011) proposed two recommendation approaches based on categories of location. The first one was a probabilistic category recommender (PCR) suggesting the category for the next destination of the active users. The other one was a probabilistic category-based location recommender (PCLR) extending PCR to recommend locations to the active users at a given time of the day. This study examined geographical and temporal influences with LBSNS data collected from Gowalla. The temporal influence model focused on finding the probability of user preferences between location category and given time differences. For example, people tend to check-in to a coffee shop at 8 am. The geographical influence model determined the home location of the user to compute probability of user preferences with location category and the given time. Zheng et al. (2011) proposed a location-history-based recommender system which used individual visiting history of users as implicit ratings on locations. These ratings were adopted to predict unvisited locations for active users. This work also presented a hierarchical-graph-based similarity measurement (HGSM). HGSM had a role in modeling each individual's location history and measuring the similarity between each user. The similarity was measured by considering three factors including 1) the mutual sequence of users' movements; 2) the popularity of visited locations; and 3) the hierarchy of geographic spaces. The work of Zheng et al. (2011) was the only work which adopted user

trajectory data to perform location suggestion. GeoLife developed by these researchers in 2009 (Zheng, Chen, Xie and Ma, 2009) was the main data source for their study.

Cheng et al. (2012) introduced a fusion framework by integrating MF with geographical and social influence for POI recommendation in LBSNSs. Cheng et al. (2012) also claimed that the geospatial influence in this work was different from the same one proposed by Ye et al. (2011). The distance of checked-in locations following a power-law distribution is proposed by Ye et al. (2011). The assumption of Cheng et al. (2012) claimed that users tend to check-in around several centers where the distance between checked-in locations and their centers followed a multi-center Gaussian model (MGM). Therefore, this work clustered the entire check-in history in database to identify the most well-known POIs as the centers. In case of social influence, probabilistic matrix factorization (PMF) was adopted to model users' preferences on locations by determining the data of the active users and their social friends. The PMF model finally was combined with MGM model to enhance the quality of recommendation. This study collected data from Gowalla for the framework evaluation. In order to predict where the users like to go next, Cheng et al. (2012a) provided other work of POI recommendation in the same year. They investigated the spatial-temporal properties of LBSNS datasets crawled from Foursquare and Gowalla. A novel MF method named FPMC-LR was proposed to incorporate the two properties. The temporal property adopted a personalized Markov Chain to offer probability of user transition. The spatial property used localized regions to constrain users' movement in order to indicate the new POIs near a user's prior check-ins. These properties eventually were conducted to combine with users'

preferences and their personalized Markov chain as a MF technique in order to predict the next destinations for users.

Urban POI-Mine (UPOI-Mine) was proposed by Ying et al. (2012). The approach had an objective to advise interesting urban POIs by mining users' preferences. This study exploited dataset from Gowalla to experiment the proposed approach. There were three factors extracted from LBSNS data consisting of social factor, individual preference, and POI popularity. The extracted features were conducted as inputs for performing data mining. Regression tree was a data mining algorithm adopted to construct a model for personalized locations prediction. They chose M5Prime as one kind of regression-trees for model construction. Nevertheless, this work did not mention to explore the influences of check-in behavior. Ignoring temporal information for POI recommendation in previous endeavors caused Yuan et al. (2013) to pick this impact to investigate. They believed that time plays a key role in POI recommendation because the users tend to visit different locations at different times in a day. Furthermore, they studied the spatial impact for location recommendation as well. User-based CF was a technique adopted to deal with a temporal influence model, while a spatial influence model was manipulated by using Bayes rule. Ultimately, they fused these two models together for performance improvement. Zhang and Chow (2013) studied user preference, social influence and geographical influence. They found that in case of geographical influence models in prior studies (Ye et al., 2011; Cheng et al., 2012), they were modeled from a common distribution for all users. Hence, the personalized geographical impact was raised as highlight in this research. It was united with user preferences and social influence in order to enhance the accuracy of POI recommendation in LBSNSs.

In order to recommend shopping places for users, Picot-Clémente and Bothorel (2013) demonstrated a method which combined three factors dominating check-in behavior. They selected Gowalla for the data source. Besides, a graphical model was raised in this study. Two researchers used the graphical model to represent the three factors comprising social graph, frequentation graph and geographic graph. They combined the three graphs into a one graph. Then they adopted the combined graph to propagate weights by using the Katz centrality method in order to select new shopping places to the users. A new problem such a cross-region CF for POI recommendation was proposed by Zheng, Jin and Li (2013). They picked the disadvantage of memory-based CF when it was required to recommend POI in a new region for users. In this work, researchers got the idea from an approach for document classification named Latent Dirichlet Allocation (LDA) to perform POI suggestion. LDA was utilized to group POIs as Topics of documents. Then it considered each user as a document and checked-in POIs of the user as the document's words. After that it represents these components as vectors before operating the recommendation with naïve CF approach.

The previous attempts as mentioned above showed the possibilities to enhance accuracy of POI recommendation by using LBSNSs data. Moreover, those studies also explore three influences including social, geographical and temporal with various recommendation techniques. Nevertheless, all of them select the native LBSNSs such as Foursquare, Gowalla, Whrrl, and Brightkite for investigation. Surprisingly, there is no mention of taking Facebook Places for testing. As mentioned earlier in this section, although Facebook Places as LBSNS of Facebook has released after those native sites, there are problems that do not appear on those work and they

could be further examined. The first is incomplete categories of attractions. Chatcharaporn et al. (2012) investigated the check-in information on Facebook Places and they found that the arbitrary checking-in of users causes some generated locations to have incomplete categories. This means that these locations cannot be suggested when users select categories of locations what they want. Thus, dealing with attractions with incomplete categories is one objective in this study. The second is exploring response time of recommendation. Because LBSNSs are online services, response time of recommendation engines should be examined. Solely accuracy of POI recommendation is evaluated in those prior attempts. The response time is adopted to measure how long the active user waits for POI recommendation. Both generating recommendation model and providing personalized results affect time usage of the systems. This study proposes the response time to measure processing time of the mobile engine based on various sizes of users' information. The third is lacking of user information. Basically, LBSNSs do not focus on profiles of users hence few attributes of profile such as name, gender, birthdate, and living area are stored in these services. Nonetheless, a user profile originally is a major element of being SNS. Implementation between the native SNS such Facebook with Facebook Places as LBSNS displays the possible way to exploit the tremendous user profile for POI recommendation. In this study, demographic filtering (DF) approach is raised to be other factors and investigating with the mobile engine. The DF approach utilizes SNS profiles of users to select friends who share similar demographic attributes with the active users before computing those friends for recommending personalized attractions. The last is unsatisfying accuracy of recommendation. Due to the large-scale data of LBSNSs and sparsity problem, the accuracy of recommendations

appeared on those previous attempts is not satisfactory. In this work, accuracy investigation is based on analyzing the effect of various data selections for operating location recommendation. The various data selections consist of using the entire data, adopting solely SNS friends and applying a DF approach to select DF-based friends. In case of recommendation operation, a user-based CF is adopted in the mobile engine in order to achieve personalized attractions recommendation.

Summary of all related work is illustrated in Table 2.6. The table merely provides significant issues about the tourism recommendation system as mentioned above. The issues include LBSNSs, data sources, recommendation approaches, influence factors analysis, evaluation as well as additional problems, respectively.

Table 2.6 Summary of related work comparison associated with a mobile engine for personalized tourist attraction recommendation using social networking services.

Topics	**Related Work**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	*****
LBSNSs													
Gowalla			✓	✓		✓	✓	✓	✓	✓	✓	✓	
Foursquare	✓	✓					✓		✓	✓			✓
Whrrl		✓											
GeoLife					✓								
Facebook Places													✓

Table 2.6 Summary of related work comparison associated with a mobile engine for personalized tourist attraction recommendation using social networking services. (Continued)

Topics	Related Work												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	*****
Data Sources													
User Profiles													✓
User Check-in Histories	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
User Trajectories					✓								
Recommendation Approaches													
Memory-based CF		✓			✓				✓	✓	✓	✓	✓
Model-based CF	✓	✓		✓				✓		✓			
Matrix Factorization-based CF			✓			✓	✓						
Influence Factor Analysis													
Geographical influence	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	
Social influence	✓	✓			✓	✓		✓		✓	✓		✓
Temporal influence				✓			✓		✓				
Demographic influence													✓

Table 2.6 Summary of related work comparison associated with a mobile engine for personalized tourist attraction recommendation using social networking services. (Continued)

Topics	**Related Work**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	*****
Evaluation													
Accuracy	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓	✓
Error Rate			✓					✓					
Response Time													✓
Additional Proposed Problems													
Missing Category of Attractions													✓

***Related Work**: 1 = Ye et al. (2010); 2 = Ye et al. (2011); 3 = Berjani and Strufe (2011); 4 = Rahimi and Wang (2011); 5 = Zheng et al. (2011); 6 = Cheng et al. (2012); 7 = Cheng et al. (2012a); 8 = Ying et al. (2012); 9 = Yuan et al. (2013); 10 = Zhang and Chow (2013); 11 = Picot-Clémente and Bothorel (2013); 12 = Zheng et al. (2013); * = This research*

CHAPTER 3

RESEARCH PROCEDURE

This chapter presents research and design issues of the mobile engine for personalized tourist attraction recommendation using SNSs. The following sections in this chapter consist of research methodology, research tools, data collection and data analysis.

3.1 Research Methodology

An approach of a design of the mobile engine for personalized tourist attraction recommendation using SNSs is adapted from the system development life cycle (SDLC) approach. Details of the design can be explained as follows:

3.1.1 Studying and analyzing the current problem of the mobile engine and related factors

The purpose of studying and analyzing the current problem is to specify both the shortcoming of the existing tourism recommendation systems and possibilities to perform continual improvement of the system. Therefore, to identify the problem, the related domains of this work are necessarily explored. These domains comprise of data acquisition and approaches of recommendation system. The first domain is the data acquisition. This research domain concentrates on using SNSs and LBSNSs as main resources for acquiring user information to perform

personalization. The other domain is detailing of tourism recommendation methodology. After identifying problems, these facts will be defined as related factors for both data acquisition and recommendation approaches. These related factors allow the mobile engine to determine the expected results. Both related factors and expected results could be exhibited as shown in Table 3.1.

Table 3.1 The related factors and expected results of a mobile engine for personalized tourist attraction recommendation using SNSs.

Related Factors	Expected Results
Data Acquisition • User profile and demographic • A list of SNS friends • Checked-in history **Recommendation Approach** • Similarity estimation of SNS Users • Personalization methodology	• Revising incomplete retrieved contents from SNSs • Recommending tourist attractions individually • Improving online response time

3.1.2 Design of the mobile engine for personalized tourist attraction recommendation using SNSs

The mobile engine for personalized tourist attraction recommendation using SNSs pays attention to the analysis of travelers' interests in SNSs prior matching those interests with attractions. Figure 3.1 illustrates a system framework of

the mobile engine, which reveals an overview of the mobile engine components and information flow underneath them.

According to the system framework, the beginning of this framework starts when an active user enters an application named “Me-Locations” on his/her mobile device. For the first time usage, the user needs to register and grant his/her permission to the mobile engine via the application. The permission allows the mobile engine to access user information on Facebook server. In order to exchange permission data, Facebook API such as Open Graph API is adopted to achieve this task with User’s Interest Acquisition module. User profiles and check-in histories as well as social relationships are fetched from the Facebook server. The social relationship in this study is considered only one level relationship. Thus, the mobile engine cannot get data belonged to friends of the active user’s friends to perform the recommendation. In case of check-in history, both check-in histories of the active user and his/her friends are fetched. When the module obtained user data from SNS server, these data may be needed to clean up, especially the category of attractions which are incomplete. A category acquisition module has an important role in retrieving additional location categories from Foursquare when there are the incomplete data from the Facebook. If Foursquare cannot provide any additional categories, those incomplete locations will be classified by labeling the proper category via a sub-module named Category Categorization. After that, the Category Categorization will send feedbacks to store in a Knowledge Base.

After storing information in the Knowledge Base, the active user is able to obtain the personalized attractions from the Recommendation sub-module. This sub-module allows the active user to configure some factors for the

recommendation including radius and categories of attractions. With radius configuration, the module needs to take the user's current location in order to perform radius calculation. To detect current location of an active user, most of the smartphones have built-in GPS. Furthermore, if smart devices cannot detect GPS signal, they have competency to adopt Wi-Fi or cell site signal to approximate the current location of a user. Hence, the module can take latitude and longitude coordinates from these approaches. Eventually, there are four parameters sent from this module to the other module named Personalized Engine for recommendation execution. The four parameters are a user ID, a current location of user, defined radius and categories of attraction.

Personalized Engine is the last module of the mobile engine. It has three main processes; filtering, matching as well as restricting. These sub-modules have a role in performing the personalization process. Those parameters are transmitted from the previous module, except the user ID, which is conducted in the last process of Personalized Engine. After finishing the personalization process, personalized attractions as results of the mobile engine will be transformed and exhibited as marks on a digital map to the active user through Me-Locations as shown in Figure 3.1.

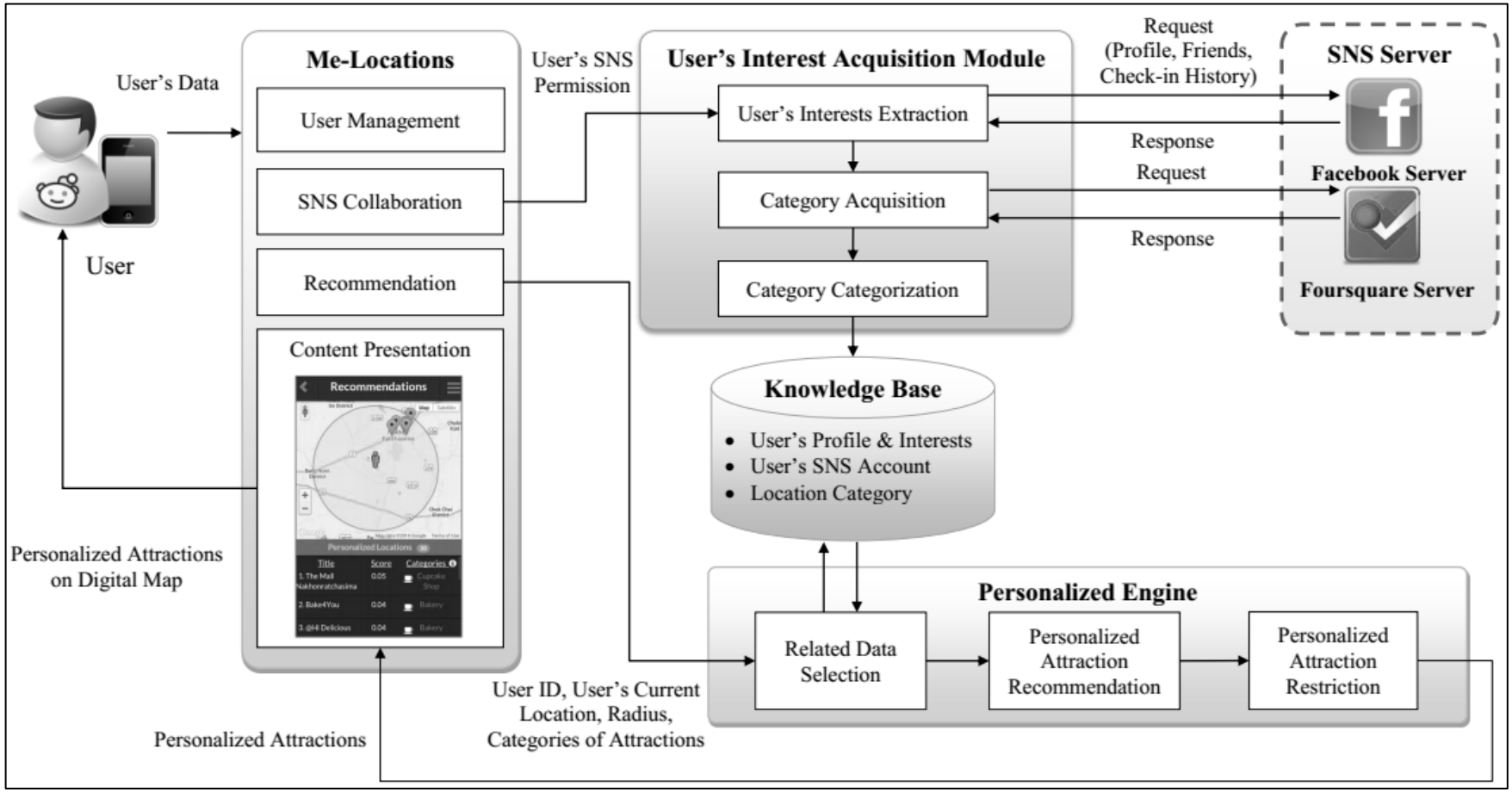


Figure 3.1 The system framework of a mobile engine for personalized tourist attraction recommendation using SNSs.

The details of components in the mobile engine can be explained as follows.

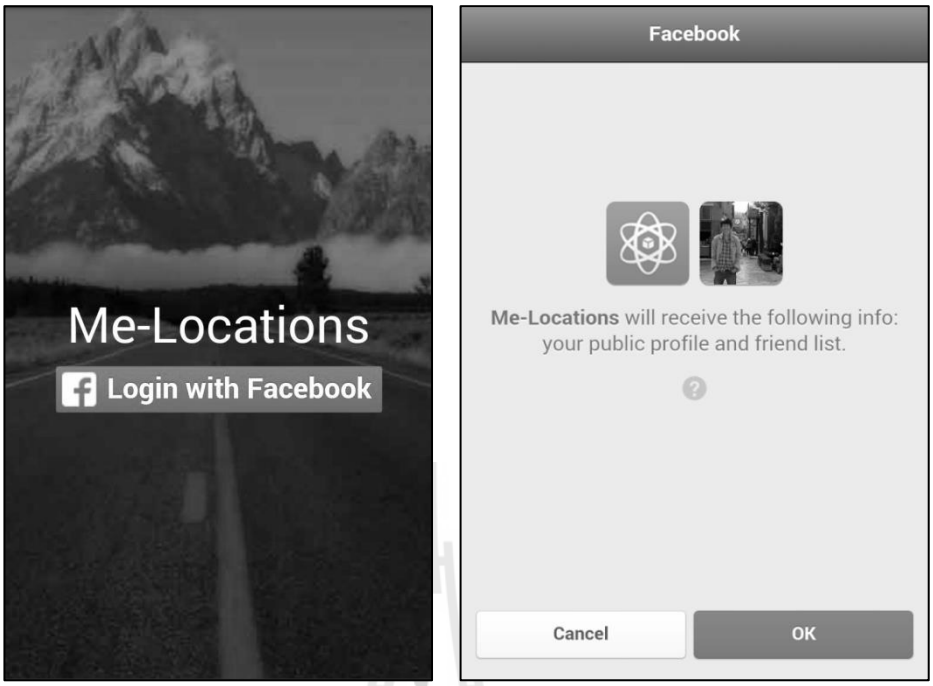
1) Me-Locations

Me-Locations application is a web-based mobile application. It needs to perform with a browser on user's smartphone. The application is employed to interact with the active user. In this study, Me-Locations development is based on HTML5 technology. It is capable to make UI of web application to be similar to the native by using Cascading Style Sheets (CSS). Furthermore, some mobile browsers have ability to add a web application icon to a mobile home screen. Address bar of browser is hidden when the users enter a web application via the exported icon. This trick makes the users to feel like they are using the native application. The capability of HTML5, CSS and JavaScript are sufficient for uncomplicated applications on mobile devices (Charland and Leroux, 2011). While the native application development needs particular SDKs to support developers, the HTML5 mobile applications do not require those standard tools for development. Thus, the phrase "write once run anywhere" is strength of HTML5. Besides, there are many frameworks which facilitate a web application to support illustration on various sizes of mobile screen such as jQuery Mobile, Bootstrap and Foundation. In addition, with the new ability of HTML5, it enables a web application to be able to retrieve information from hardware, especially geographical information from GPS. However, the interaction of HTML5 applications is quite slow to response the user when compared with the native application. Using the application with a browser is another disadvantage of HTML5 implementation on mobile devices.

In this research, Me-Locations is designed based on the HTML5 technology because it facilitates testing the mobile engine with browsers on different mobile operating systems, especially presentation. The application has four sub-modules consists of User Management, SNS Collaboration, Recommendation, and Content Presentation. The detail of each sub-module can be described as follows:

1.1) User Management

When active users enter Me-Locations, the first sub-module that they contact with is User Management. This sub-module composes the user interface in order to either register or login. In case of the new users, the registration is performed by storing users' accounts into the system. Then, it obtains the authorization from the active users in order to reach their information on SNSs. The user interfaces (UI) of registration and authorization can be illustrated in Figure 3.2(a) and Figure 3.2(b), respectively. In order to achieve the system registration, users need to press the Facebook login button. After that the users are required to authorize the application in order to access their SNS information. Finally, the completion of authorization will lead the users to a home screen of Me-Locations as shown in Figure 3.3. The home screen is related to the next two sub-modules of Me-Locations named Social Networking Services (SNSs) Collaboration and Recommendation. If the active users already perform the registration, they are capable to interact with the Recommendation sub-module in order to obtain the personalized attractions immediately.



(a) (b)

Figure 3.2 (a) The system registration screen. (b) The Facebook application authorization.

Figure 3.3 exhibits the home screen of Me-Locations. The home screen consists of a title bar with an option menu, a profile picture of active user, an active user's name and current location, the active user's statistics, and two tabs of Recommendation and SNS Collaboration sub-modules. The option menu contains Terms and Policies of this application and a logout function. With Google Map API implementation, the current location of active user can be transformed into address. Next, there are three statistics of active user below the address adopted to furnish the number of friends, check-ins and locations. The active user is able to touch these

numbers to see their details. The two tabs under the statistics are related to the following sub-modules.

Figure 3.3 The home screen of Me-Locations.

1.2) Social Networking Service (SNS) Collaboration

With the first time usage, the active user is notified to enter the second tab of Me-Locations because the mobile engine cannot provide any recommendation without the active user's SNS information. Consequently, the second tab has a role in interacting with the active user for SNS information update and retrieval. Figure 3.4 presents the user interface of SNS Collaboration. This sub-module acts as a bridge between users and Facebook server. The authorization from the first time usage enables the mobile engine to be able to access the active user's

information on SNS server. With this tab, the active user can update both SNS information of his/her own and his/her friends. As shown in Figure 3.4, there are two parts of updating SNS information. The first one is updating SNS information of the active user. Another one is adopted to update SNS friends' information of the active user. When the user touched either part, the sub-module will continue its process with the User's Interest Acquisition module in order to retrieve users' information from Facebook server.

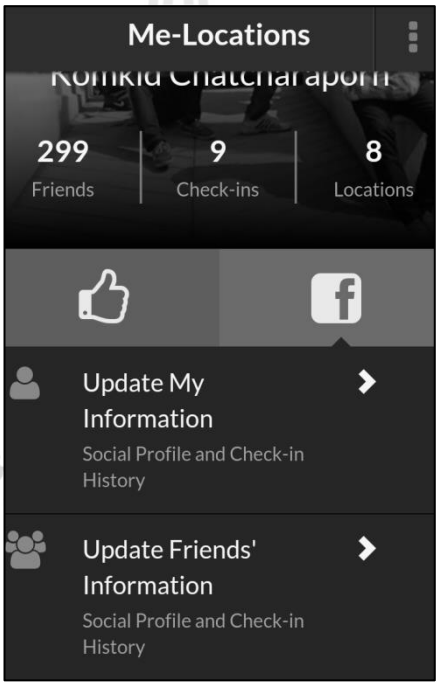


Figure 3.4 The user interface of SNS Collaboration sub-module.

1.3) Recommendation

As illustrated in Figure 3.5, the first tab presents a user interface of the Recommendation sub-module. It lets the active user to assign distance of

radius and select categories of attractions. Unit of the distance is kilometer and its maximum is 100 kilometers. The application uses latitude and longitude of the active user as a central point. The central point facilitates the mobile engine to limit the number of recommended attractions by determining radius around the current point of the active user. For example, 10 kilometers around the active user's current location. In case of category selection, there are 11 categories adapted from QALL-ME ontology (Ou, Pekar, Orasan, Spurk, and Negri, 2008). To obtain suggestion results, the active user solely touches a button named Recommend Me. Personalized Engine module receives requests from this sub-module in order to operate recommendation and send the outcomes back to display at Content Presentation sub-module.

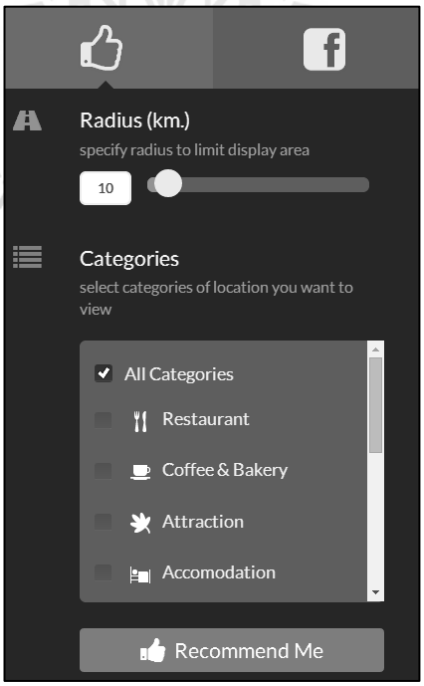
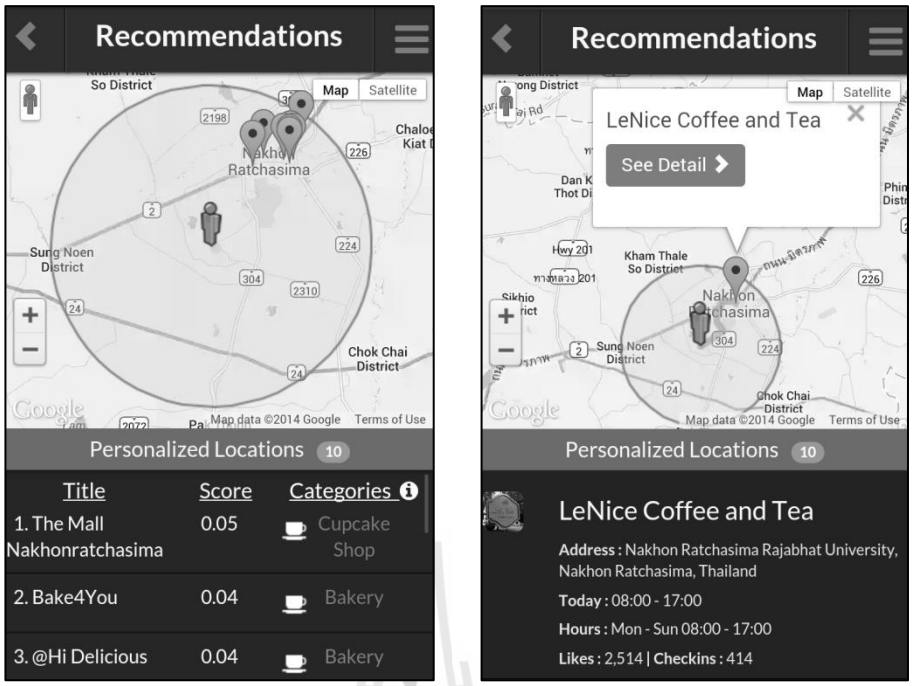


Figure 3.5 The user interface of recommendation sub-module.

1.4) Content Presentation

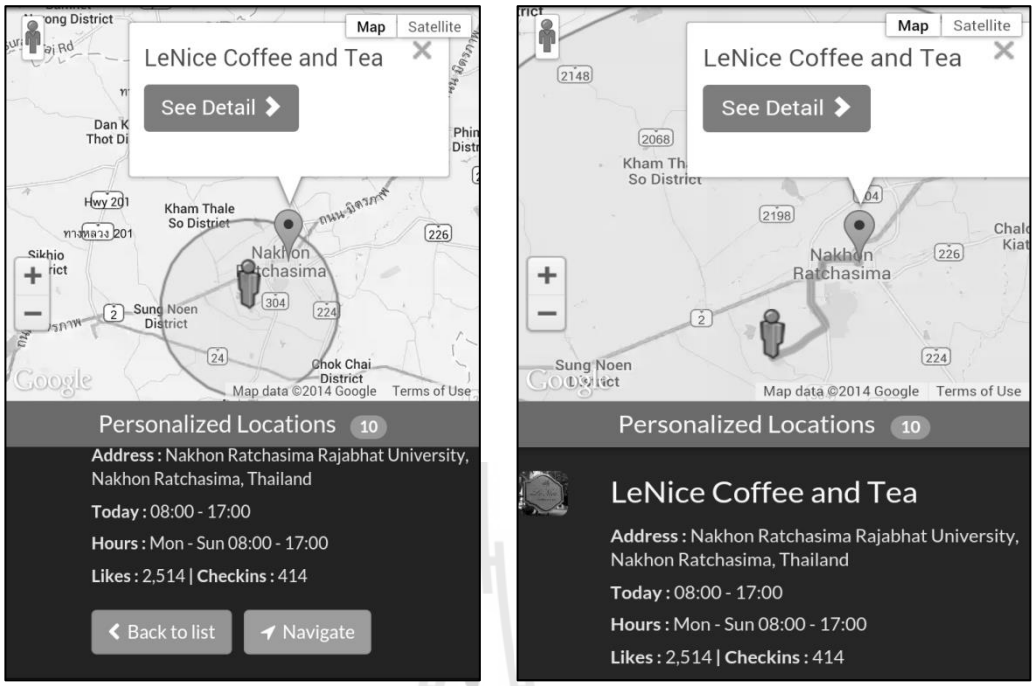
The last sub-module is Content Presentation. Me-Locations does not only receive requests from active users but also displays the responses to them. The responses are the personalized attractions in the form of marks on a digital map. An example of displaying personalized attractions on a digital map, called map perspective, is shown in Figure 3.6(a). A list of personalized places is shown beneath the digital map. Titles of the list indicate the number of recommended attractions. In the list, there are three results of recommendation comprising titles of attractions, recommending scores as well as categories of attractions. In order to view details of an attraction, the active user either touches on a red marker on the map or taps a title of desired attraction in the list then touches the See Detail button. Figure 3.6(b) introduces the details of attraction.



(a) (b)

Figure 3.6 (a) The presentation of the mobile engine output in map perspective. (b) The details of attraction.

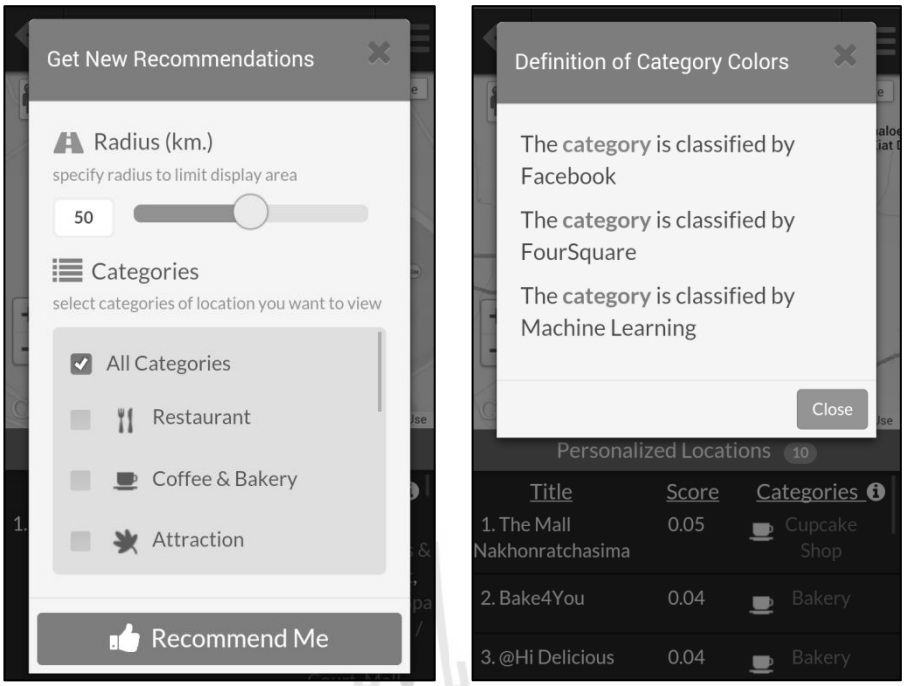
In addition to viewing details of attractions, the active user can get a direction to the selected attraction by tapping a Navigate button as displayed in Figure 3.7(a). The consequence of navigation is notified as shown in Figure 3.7(b).



(a) (b)

Figure 3.7 (a) A button for attraction navigation. (b) The result of navigation.

The title bar on the top of Content Presentation screen has two buttons. The left one is a back button used to go back to the home screen. Another one is an option button. It is adopted to perform re-recommendation as exposed in Figure 3.8(a). The option allows the active user to obtain new results of suggestion by choosing new configurations. Notice that categories of attraction displayed in the list have many text colors. The definition of these colors can be explained by touching the third column title of the list named “Categories”. The difference of colors is based on different category acquisition methods as shown in Figure 3.8(b). Details of the category acquisition will be described in the following module.



(a) (b)

Figure 3.8 (a) The user interface of getting new recommendation result. (b) The definition of each category colors.

2) User's Interest Acquisition Module

After the active user presses one of the updating buttons from Me-Locations, the mobile engine will send requests for SNS information retrieval to Facebook servers. User's Interest Acquisition module has duty to perform this action. As mentioned in the beginning of section 3.1, this study focuses on the user's interests in a tourism domain. Hence, some user data are extracted from the SNS such as profiles, check-in histories and social relationships. Profiles and check-in histories belong to the active users and their friends in SNS. This module consists of three sub-modules. The first one is User's Interests Extraction, which is employed to fetch the

user interests. The second one is Category Acquisition. It has a role in getting additional categories of locations from Foursquare based on pre-fetched locations from Facebook. In case of Foursquare cannot provide any additional categories, the last one named Category Categorization is operated to revise the incomplete data obtained from Facebook, particularly the category of checked-in attractions by implementing machine learning with latent semantic analysis.

2.1) User's Interests Extraction

The first sub-module has a role in fetching the active user's interests by forwarding requests to Facebook servers through the Open Graph API (Ko, Cheek, and Shehab, 2010). Then, Facebook server will send responses back to this sub-module in JSON format (Chuang, Lin, Ren, and Yeh, 2011). Even though, there are numerous interests of users from the Facebook but this study only concentrates on the interests in the tourism domain. There are three SNS data extracted from Facebook servers. The three data are a profile of active user and his/her friends, one level social relationship and one check-in history. The profile data includes name, gender, relationship status, living location, and education information. The social relationship is friends of active user who have check-in histories on Facebook Places. The last data is the check-in history of active user and his/her SNS friends. The history transaction consists of IDs of place and timestamp of check-in. Place ID can be adopted to fetch additional information of place such as name, description, category, latitude and longitude coordinates, the number of check-ins, and tagged data. Extracting data for each active user may take a long time to complete because each of them has plenty of friends. Hence, multithreaded programing (MP) proposed by Chatcharaporn, Angskun, and Angskun (2013) is

adopted to improve speed of data extraction. This approach divides friends' data of active user into *n*-set equally before distributing datasets to perform extraction simultaneously with *n*-worker pages. After obtaining all responses, the sub-module converts them from the JSON format to the array format and sends them to store in the knowledge base. In case of attractions with missing categories, this module manipulates them by sending to the next sub-module named a Category Acquisition module.

In order to access the user's information in Facebook server, there are regulations which developers must follow on the Facebook policies. For example, the developer needs to build the Facebook applications as middleman for cooperating information transfer between Facebook server and developer server. Another example is the user who wants to register Facebook applications must authorize permissions to the applications. Although, creating applications is performed on Facebook service, but all scripts for running the applications are set up in developer servers. Therefore, forwarding requests to fetch the users' interest information comes from the developer servers. The cooperation between the two servers is executed through the Open Graph API. The API currently becomes de facto standard API for Facebook application developers (Ko et al., 2010).

2.2) Category Acquisition

The second sub-module is activated when the prior sub-module has already sent incomplete results to it. The categories of locations are necessary because lacking of them could mislead the active users when they make a decision to visit those places. A Foursquare application is also required to act as middleware

collaborating between the Foursquare server and the developer server via Foursquare API. Hence in this sub-module, there is a Foursquare application. Although, there are many attributes of fetched attractions from Facebook but only three of them are selected as parameters for Foursquare API. The three parameters comprise location's name, latitude and longitude coordinates. These parameters will be attached with Foursquare application's ID and Secret Code when the sub-module makes a request to the Foursquare server. After that the Foursquare server returns the responses back in the form of JSON. The responses include place's detail such as ID, name, contact, category, location, and the number of checking-ins. Then, the JSON format will be converted to the array format. Finally, the incomplete categories of user's checked-in places fetched from Facebook will be replaced by the additional categories and stored in the Knowledge Base. However, there are some places which the Foursquare cannot provide the additional categories. Hence, there is the last sub-module operating to classify those places.

2.3) Category Categorization

The last sub-module has roles in categorizing location information from the previous sub-module. The categorization labels the most appropriate categories for each of user's checked-in attractions. These places could be related to multiple or no categories and some of them are incomplete. In this research, latent semantic analysis (LSA) and machine learning (ML) techniques are adopted to operate text categorization. Text categorization is a classification technique in data mining. It is able to automatically assign natural language texts based on their content to predefined classes or categories. The mobile engine takes

this technique to categorize attractions with missing categories by determining their title. This study adopts the approach proposed by (Chatcharaporn et al., 2014) to implement the text categorization. There are five main processes as follows: (1) Data Collection, (2) Data Pre-processing, (3) LSA Implementation, (4) Mapping Semantic Space with Categories, and (5) Attraction Categorization. All steps of the five main processes can be illustrated in Figure 3.9.

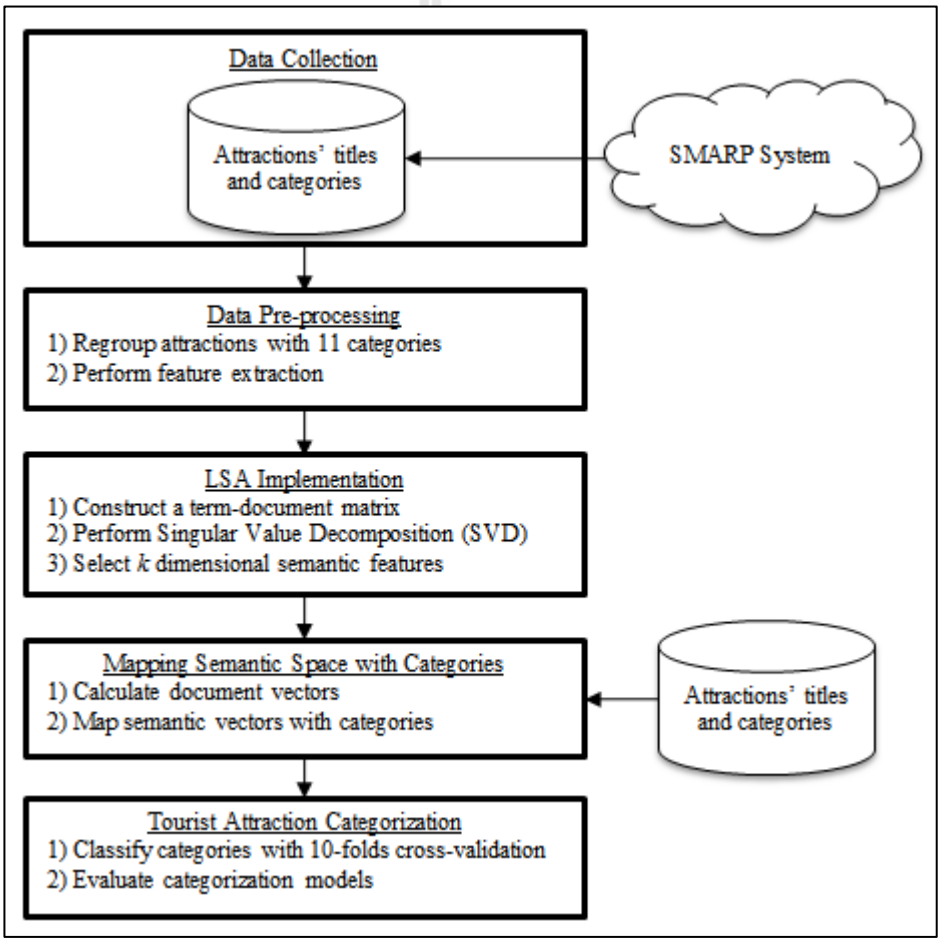


Figure 3.9 A framework for categorizing tourist attractions using LSA and ML techniques.

2.3.1) Data Collection

The first process has duty to collect the related data by retrieving them from an online system, named SMARP (Social network in Mobile Augmented Reality for Personalization) proposed by Chatcharaporn, Angskun, and Angskun (2011). The related data for text categorization is attractions which have both titles and categories. The categories of each attraction are not only taken from Facebook but also fetched from FourSquare because some attractions are not assigned categories by Facebook users. Furthermore, the system also stores information of some locations in Thailand taken from LonelyPlanet.com. In SMARP's database, there are 28,536 attractions. Their titles are written in many languages, such as Thai, English, Korea, Chinese, and Japanese. This study mainly focuses on the attractions written in English language. In other words, the proposed module has a capability to categorize attractions based on English language only. There are 11,374 attractions would be extracted from the system. The name of extracted attractions is unique and each of them has its own categories. Obtaining these attractions from the SMARP system can be performed by implementing a PHP program. After that, the retrieved attractions will be stored in the knowledge base and adopted in the next process by using MySQL and a PHP language.

2.3.2) Data Pre-processing

After collecting attraction data, two steps are proposed to pre-process the collected data. The first step is regrouping categories of the collected attractions. Using the large number of categories could lead the categorization model to provide poor results of prediction because most categories of attractions occur very

infrequently. Hence, regrouping those categories makes the number of them to be smaller and to increase the frequency of the occurrence. The category regrouping method is able to improve performance of the categorization model. The second step is extracting features from attractions' titles. Each of features or terms represents a word extracted from the titles. These features are used to construct a term-document matrix in the next main process. The benefits of feature extraction are to keep the significant features and get rid of insignificant features from the collected data. Therefore, this data pre-processing facilitates a machine to reduce both time and memory usage when constructing the categorization model. Besides, it enables the model to provide more accurate outputs.

1) Regrouping categories of the collected attractions

As mentioned above, all the collected attractions are tagged by at least one category. Those categories of each attraction could be obtained from Facebook, FourSquare and LonelyPlanet. These resources have plenty of unique categories. Facebook, FourSquare and LonelyPlanet have 161, 389 and 13 unique categories, respectively. Thus, the total number of unique categories is 563. As previously mentioned, taking all categories to perform the categorization could lead to unsatisfied results. To deal with this situation, regrouping categories is required in order to group some of them which are similar in the sematic way. For instance, cafeteria, cafe, coffee shop, dessert shop, ice cream shop, tea room as well as bakery are capable to be grouped into "Coffee & bakery" as a unique category. A tourism ontology named QALL-ME is taken to regroup the original categories of the 11,374 attractions. There are 11 categories of QALL-ME ontology chosen to perform this task. The 11 categories

include accommodation, attraction, coffee and bakery, convention exhibition, entertainment, health, public, restaurant, shopping, sports, and terminal. After regrouping, there are 1,124 attractions that cannot be regrouped because their original categories cannot be mapped with the 11 categories of QALL-ME ontology, therefore these attractions are not taken to build the categorization model. Hence, only 10,250 attractions of collected data are adopted to construct the categorization model. Table 3.2 shows the number of attractions in each of the 11 categories.

Table 3.2 The number of attractions in each category.

Category of attractions	**Number of attractions**
Accommodation	1,380
Attraction	1,155
Coffee & bakery	1,046
Convention exhibition	50
Entertainment	308
Health	255
Public	100
Restaurant	4,161
Shopping	983
Sports	250
Terminal	562

2) Extracting features from attractions' titles

The purpose of feature extraction is to extract significant terms or features from text documents. Feature extraction is a useful technique in order to decrease dimensionality and get rid of noises from documents. In this research, text documents are titles of attractions and features are a set of words extracted from the titles. For example, "china town boston" is an example of a text document and the extracted features of this text document are "china", "town" and "boston".

The following explanations are the four steps of feature extraction in details:

2.1) Text Cleaning: All titles of attractions are transformed into plain text by removing non-alphabetical characters and converting text to lower case.

2.2) Words Segmentation: In this step, terms are split from the plain text by considering spaces between them as a separator.

2.3) Stop Words Removal: Some of separated terms, which are not significant, will be discarded, especially article, preposition, pronoun, and conjunction. These kinds of words are called stop words. The examples of stop words include "a", "an", "the", "he", "she", "they", "in", "on", "and", "who", "which", and "that".

2.4) Stemming Word: Finding a root form of terms without prefixes and suffixes is the last process in feature extraction. For instance, the word "guest house" can appear in different forms in many documents, such as "guest house", "guest houses", and "guest's house". This technique is able to reduce

frequency of these features which occurs in documents by transforming these features into a unique word as “guest house”. Porter’s Stemming algorithm (Porter, 1980) based on a PHP program is used in this step because it is a suitable algorithm for stemming English language. After stemming, the redundant stemmed-words are removed and the number of existences is 6,787 words.

2.3.3) LSA Implementation

Latent Semantic Analysis (LSA) is an information retrieval technique proposed by Dumis, Fumas, Landauer, Deerwester, and Harshman (1988). Currently, it is the famous technique in text categorization (Yu, Xu, and Li, 2008; Loni, Khoshnevis, and Wiggers, 2011). The basic idea of LSA is using statistic and linear algebra in order to project the high dimensional document vectors into the low dimensional latent semantic space (Lv and Liu, 2005). Dimensionality reduction using the LSA is derived by singular value decomposition (SVD). According to Huang (2011), LSA process generally consists of three steps: 1) Constructing a term-document matrix; 2) Projecting the term-document matrix into latent semantic space using SVD; and 3) Reducing the latent semantic space.

1) Constructing a term-document matrix

In the first step, the term-document matrix A is constructed to represent the relationship between terms and documents. Basically, the rows of matrix A represent terms and the columns represent documents. The matrix A comprises m terms and n documents (m x n matrix). The cells of matrix A contain weights of terms in documents, which indicate the significant of a term in a document.

Size of the term-document matrix A in this work is 6,787 x 10,250. In the matrix, each row represents a unique stemmed-term and each column represents a title of attraction. Each cell of the matrix contains binary values 1 and 0 that express an occurrence of a specific term in a specific attraction's title. Table 3.3 demonstrates details of a term-document matrix *A*. An example of attractions' titles and terms in Table 3.3 is as follows: "imperial queen park hotel bangkok", "queen sirikit nation covent center" and "hua hin resort" representing the $Title_1$, $Title_2$ and $Title_3$; while "center" and "queen" represent the $Term_1$ and $Term_2$.

Table 3.3 A term-document matrix.

Terms	**Titles of Attractions**				
	$Title_1$	$Title_2$	$Title_3$	…	$Title_{10,250}$
$Term_1$	0	1	0	…	0
$Term_2$	1	1	0	…	1
…	…	…	…	…	…
$Term_{6,787}$	1	0	1	…	1

2) Projecting the term-document matrix onto latent semantic space using SVD

The second step is performing an SVD algorithm using Python with its plugins. The objective of SVD execution is to project the term-document matrix *A* onto a latent semantic space. The semantic space presents a semantic value of relationship between terms and attractions' titles. The SVD is a standard decomposition technique manipulated to compute the singular value in

linear algebra. The matrix A is decomposed into three matrices after accomplishing the SVD as shown in an Equation 3.1.

$$A = USV^T \tag{3.1}$$

where U and V^T are orthogonal matrices and their columns contain eigenvectors of AA^T and A^TA, respectively. S is a diagonal matrix consisting of the eigenvalues of AA^T in the diagonal sorted in a descending order. The composition of the three matrices can be illustrated in Figure 3.10.

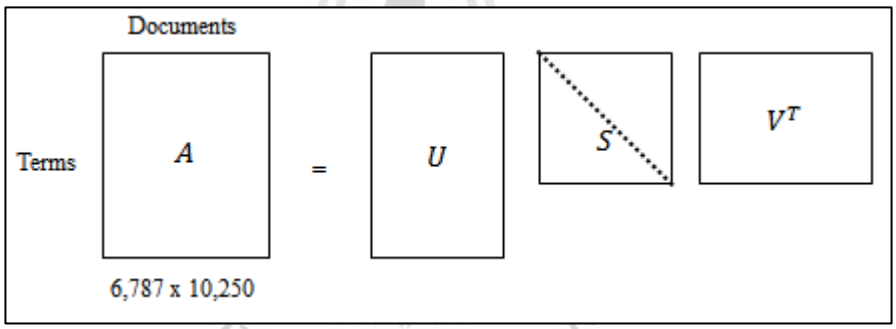


Figure 3.10 Decomposing a term-document matrix A by using SVD.

3) Reducing The Latent Semantic Space

The last step is reducing the semantic space. This process is also called dimensionality reduction. The main purpose of reduction is removing the noises from the semantic space. Moreover, the process also reduces the amount of data and memory usage. Decreasing the semantic space can be performed by selecting the k largest singular values of three matrices obtained from the SVD

process. As displayed in Figure 3.11, the first *k* columns of matrix *U* are chosen as U_k. Then the first *k* rows of matrix V^T are selected as V_k^T. Finally, the first *k* factors of the diagonal elements are selected as S_k.

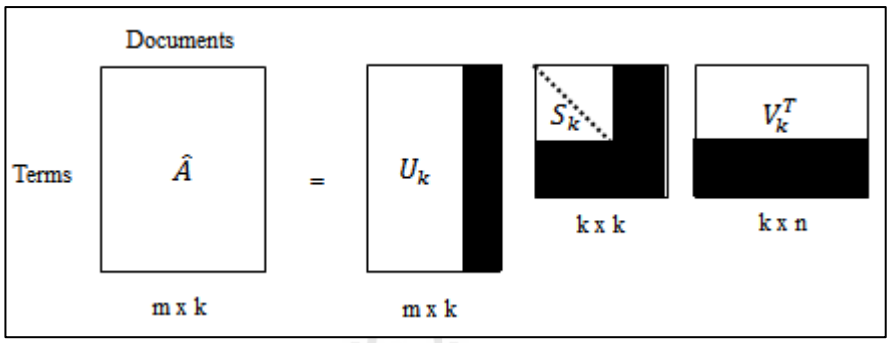


Figure 3.11 Selecting *k*-dimensional spaces of the three matrices.

The product of the three reduced matrices provides a matrix $\hat{A}$. The matrix $\hat{A}$ is only approximately equal to the matrix *A* of any rank-*k* matrix in the least square sense. The Equation 3.2 demonstrates the approximation of matrix *A* as an Equation 3.2.

$$A \approx \hat{A} = U_k S_k V_k^T \tag{3.2}$$

In this work, U_k and S_k are only two matrices adopted in the next process in order to map 10,250 locations as an original dataset to latent sematic space.

2.3.4) Mapping Semantic Space with Categories

The two matrices U_k and S_k as the output from a prior main procedure are adopted to perform in this procedure. The procedure consists of two steps: 1) Calculating document vectors; and 2) Mapping semantic vectors with categories. The first step mainly focuses on transforming document vectors of 10,250 attractions to semantic vectors. The second step has a role in mapping 10,250 attractions to the original categories after they are transformed into the semantic vectors.

1) Calculating document vectors

The 10,250 attractions are conducted in forms of document vectors with *k*-dimensional space as same as the collection of documents. Each vector contains values of a term related with each document as defined in an Equation 3.3.

$$d_j = \lfloor w_{j1}, w_{j2}, ..., w_{j6787} \rfloor \tag{3.3}$$

where w_{ji} is weight of *i*-th indexing terms in the document *j*. In this work, the weighting value is binary values 1 and 0.

All document vectors are projected onto sematic vectors by using the Equation 3.4.

$$\hat{d} = d^T U_k S_k^{-1} \tag{3.4}$$

2) Mapping semantic vectors with categories

After that the sematic vectors will be mapped with their original category. The category is added as the last value of each sematic vector as shown in an Equation 3.5.

$$\hat{d}_j = \left[w_{j1}, w_{j2}, ..., w_{jk}, \#category \right] \tag{3.5}$$

2.3.5) Tourist Attraction Categorization

1) The set of latent feature vectors is employed to construct categorization models. The models have a role in categorizing types of attractions. The widely used algorithms which are implemented with LSA include Naïve Bayes (NB) (Lv and Liu, 2005; Wan, and Tong, 2008; Inrak, and Sinthupinyo, 2010), Decision Tree (J48) (Inrak, and Sinthupinyo, 2010), Back-Propagation Neural Networks (BPNN) (Yu et al., 2008; Loni et al., 2011) as well as Support Vector Machine (Yu et al., 2008; Lv and Lui, 2005; Inrak, and Sinthupinyo, 2010; Hillard, 1996). In this experiment, the performance comparison of categorization models is based on various dimensions of features and the four mentioned ML algorithms. The purpose of comparison is to find the most appropriate model for categorizing tourist attractions. The following explanation is details of each proposed algorithm:

a) Naïve Bayes

The Naïve Bayes (NB) has been widely used for text categorization. The NB adopts the probability to categorize documents based on Bayes' theorem (McCallum, and Nigam, 1998). Even though NB is a very simple algorithm, it is very effective. The NB was proposed for text categorization by Lewis

in 1998. The Equation 3.6 and 3.7 describe the ways of NB algorithm in order to categorize attractions.

$$P(Category_j \mid Attraction_i) = \frac{P(Category_j) \times P(Attraction_i \mid Category_j)}{P(Attraction_i)} \quad (3.6)$$

where $P(Category_j \mid Attraction_i)$ is the probability that the $Attraction_i$ belongs to the $Category_j$. The $P(Category_j)$ is the probability of a given $Category_j$. The $P(Attraction_i)$ is the probability of a given $Attraction_i$. The $P(Attraction_i \mid Category_j)$ is the probability that the $Attraction_i$ is in $Category_j$. It is calculated from the probability that is given the set of features in $Attraction_i$ which occurs in $Category_j$. The calculation can be displayed in the Equation 3.7.

$$P(Attraction_i \mid Category_j) = P(f_1, f_2, ..., f_n \mid Category_j) = \prod_{k=1}^{n} (f_k \mid Category_j) \quad (3.7)$$

b) Decision Tree

The Decision Tree is a famous algorithm implemented for the categorization task. The well-known categorization algorithms are based on the decision tree, such as ID3 and J48 (Quinlan, 1986). Categorization models adopted with these algorithms are in the form of tree-shaped structures. The structures consist of nodes and branches which represent a set of rules for categorization. The difference between tree-based algorithms is selecting tree's nodes. ID3 takes values of information gain or entropy to select attributes. The

attributes which provide the highest value of information gain or the lowest value of entropy are decided as the nodes. In case of J48, it adopts both information gain and entropy to choose the nodes. Furthermore, it also takes Gain Ratio to select the nodes as well. In this work, J48 is selected as a representative of decision tree algorithm.

c) Back-Propagation Neural Networks

The Back-Propagation Neural Networks (BPNN) is a famous algorithm of artificial neural networks. It is extensively used in text categorization because it can be adopted with both linear and non-linear problems (Yu et al., 2008; Loni et al., 2011). Besides, it has a capability to provide good results of categorization. Generally, the BPNN consists of at least three layers, including one input layer, at least one hidden layer, and one output layer as shown in Figure 3.12. Initially, the inputs will be propagated through the network in order to get the responses of the output layer. After that, the feedbacks are sent backward to decrease errors. In this process, weights in all hidden layers are adjusted. While the propagation is processing, the weights are revised repeatedly. This process makes the results of the output to be enhanced.

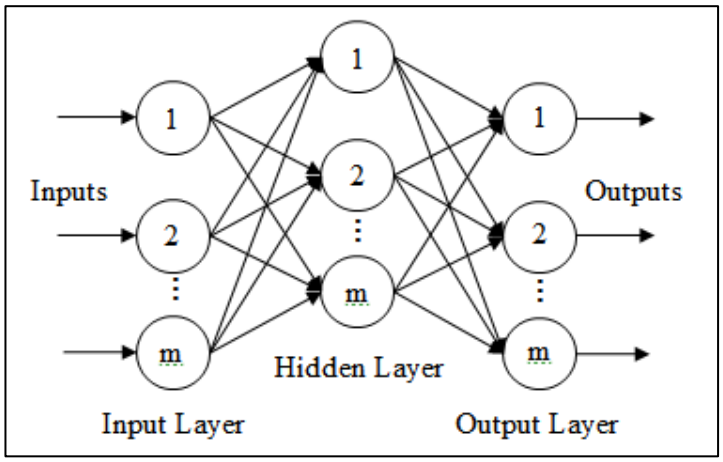


Figure 3.12 A typical structure of BPNN.

d) Support Vector Machine

The Support Vector Machine (SVM) is a supervised ML technique. It was introduced by Vapnik (1995). The basic idea of SVM is to locate the best possible surface in order to partition a dataset by a linear equation. There are two main steps performed in SVM classification. The first step is mapping the non-linear data into a high dimensional space using a kernel function. This space is also called feature space. The second step is creating a hyperplane employed to separate the data in feature space into two sets with the maximum margin. An example of SVM categorization is demonstrated in Figure 3.13. In a domain of text categorization, SVM is very famous and has been proven to be the best algorithm. However, the disadvantage of this algorithm is selecting an appropriate kernel function to data. In this work, SVM is performed using a polynomial kernel as the kernel function.

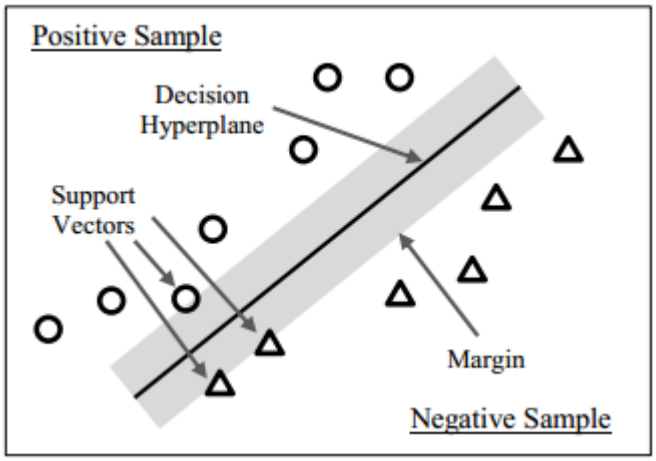


Figure 3.13 An example of SVM classification.

2) The constructed models are trained and tested with Weka (Hall, Frank, Holmes, Pfahringer, Reutemann, and Witten, 2009). The standard 10-fold cross-validation is implemented for training and testing models. Recall rate as an information retrieval measure is considered to indicate the performance of categorization models. The evaluation detail of attraction categorization is demonstrated in Chapter 4. The most appropriate approach from the comparison is considered to be applied in the mobile engine. With the approach implementation, all categorized results are stored in the Knowledge Base and the following module can take them to operate personalized recommendation.

3) Personalized Engine

The last module of mobile engine is Personalized Engine which plays an important role in providing personalized attractions to the active user. This module is activated to execute suggestion when there is a request sent from

Recommendation sub-module of Me-Locations. This module has three main processes performed to accomplish personalization. The three processes consisting of related data selection, personalized attraction recommendation, and personalized attraction restriction and their detail can be described as follows.

3.1) Related Data Selection

The first process of Personalized Engine starts with fetching related data from the knowledge base. The related data comprise users, check-in histories and attractions. These data eventually are taken to operate recommendation in the next process. Nevertheless, choosing data is a very significant issue because it affects the quality of suggestion. In this research, the users who have check-ins greater than or equal to five attractions are determined. Furthermore, to seek the best approach for personalized suggestion, there is a comparison between various data selections. Three data selections are proposed for the comparison. The first one is using the entire users in the knowledge base for finding friends of the active user. These users are considered as friends, if they are similar to the active user. The process of similarity analysis is discussed in the next process. The second one is considering only SNS friends. This method solely chooses SNS friends of the active user to perform recommendation. The last one is employing a demographic filtering (DF) approach in order to decrease the number of users before operating advice.

Basically, a DF technique applies profiles of users to estimate likeness between them. This research adopts SNS profile of users such as gender, relationship status, living location as well as educational information to find DF-based similarity. Similarity estimation is computed by using Cosine similarity

measure. The users who have DF-based similarity more than 0.0 are selected as friends of the active user. Table 3.4 and Equation 3.8 demonstrate the way to achieve -Cosine similarity with the DF approach.

Table 3.4 Examples of demographic vectors.

Users	Gender		Relationship Status		Living Locations		Educational Institutes	
	Male	Female	Single	Married	Bangkok	Nakhon Ratchasima	Suranaree University of Technology	Kasetsart University
1	1	0	1	0	0	1	1	0
2	1	0	1	0	1	0	1	1
3	0	1	0	1	1	0	0	1

Table 3.4 presents demographic features of three users. Binary values are adopted to indicate an association between users and demographic attributes. The number 1 represents the demographic attribute belonged to the user and the number 0 is otherwise. Before calculating Cosine similarity, the demographic attributes need to be transformed into vectors as inputs. For example, the set [1,0,1,0,0,1,1,0] is a demographic vector of the first user. Moreover, to compute Cosine similarity, the input vectors should always be the same length. Thus, initially the demographic attributes of two users should be merged for the equal length. For instance, {male, single, Nakhon Ratchasima, Suranaree University of Technology} is a feature set of the first user, {male, single, Bangkok, Suranaree University of Technology, Kasetsart University} is a feature set of the second user and the consequence of combination is {male, single, Nakhon Ratchasima, Bangkok, Suranaree University of Technology, Kasetsart University}. Lastly, generating new

input vectors for Cosine similarity computation is based on the combined set. In this study, the notion of attribute combination is proposed to optimize computation of Cosine similarity. Because using all attributes of demographic to produce the similarity spends a long time to be accomplished. The optimization could relieve processing time of Cosine calculation by creating input vectors with particular length for each pair of users. This means that the similarity computation between the first user and the second user might have the length of vectors greater or less than the computation with other users.

The following example of Cosine similarity computation relies on two vectors of the first user and the second user as shown in Table 3.4. The two vectors are employed to find likeness between the two users. The Equation of Cosine similarity is demonstrated in the Equation 3.8.

$$cosine(x, y) = \frac{x \cdot y}{\|x\| \times \|y\|} \tag{3.8}$$

where $\cdot$ represents the dot product, $\|x\|$ represents the length of vector *x*, and $\|y\|$ represents the length of vector *y*. The length of a vector can be defined as the Equation 3.9.

$$\|x\| = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{3.9}$$

With this example, the two vectors are

$$x = (1,0,1,0,0,1,1,0)$$

$$y = (1,0,1,0,1,0,1,1)$$

Then

$$\|x\| = \sqrt{1^2 + 0^2 + 1^2 + 0^2 + 0^2 + 1^2 + 1^2 + 0^2} = \sqrt{4} = 2$$

$$\|y\| = \sqrt{1^2 + 0^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2 + 1^2} = \sqrt{5} = 2.236$$

The result of dot product is

$$x \cdot y = (1 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times 1) + (1 \times 1) + (0 \times 1) = 3$$

Hence, the Cosine similarity is

$$cosine(x, y) = \frac{3}{2 \times 2.236} = \frac{3}{4.472} = 0.67$$

The value 0.67 is a similarity value between the first user and the second user measured by using Cosine similarity. The cosine similarity of two users will range from 0 to 1 where 1 indicates perfect similarity and 0 indicates they are not similar.

Eventually, all users picked by the three data selections are determined as friends of the active user. The next process of Personalized Engine takes these friends to implement recommendation. The main task of the next process is to find 15-nearest neighbors who share similar interests in a tourism domain with the active user and use their check-in histories to deliver personalized attractions.

3.2) Personalized Attraction Recommendation

The second process of Personalized Engine is Personalized Attraction Recommendation. It has a role in performing personalized recommendation to the active users. After the data selections have finished, all associated data, particularly check-in histories of the active user and the selected friends, are adopted in this process to operate suggestion. Even though there are several techniques applied for attraction recommendation, a user-based collaborative filtering (CF) technique is proposed in this study because it is well implemented for the location suggestion (Ye et al., 2011). The user-based CF approach is a memory-based algorithm. This technique seeks the other users who share the similar check-in behavior with the active user to achieve recommendation. With advantage of the memory-based algorithm, the user-based CF approach has capability to acquire the recent data in order to perform recommendation. This advantage enables the active user to obtain novel suggested results.

The following procedures of Personalized Attraction Recommendation are based on the user-based CF approach including check-in representation, similarity estimation as well as recommendation. The details of three procedures can be exhibited as follows:

3.2.1) Check-in Representation

To represent check-in behavior of users, the first procedure presents a user-attraction matrix as displayed in Figure 3.14. The check-in histories of the active user and the chosen friends are converted into this matrix. The matrix consists of *m* users and *n* attractions. In the user-attraction matrix, each cell $C_{u,a}$ corresponds to the check-in history of a user *u* at an attraction *a* where the value

$C_{u,a}$ = 1 indicates the user *u* has checked-in the attraction *a* and $C_{u,a}$ = 0 is otherwise. Finally, this matrix is employed as input of similarity estimation and recommendation.

		Attractions				
		1	2	3	…	n
Users	1	1	1	0	0	1
	2	1	0	0	1	0
	3	0	0	0	1	1
	⋮	0	0	0	0	1
	m	1	0	0	1	1

Figure 3.14 A user-attraction matrix.

3.2.2) Similarity Estimation

The user-attraction matrix generated from the previous procedure is used as input in similarity estimation. The objective of the second procedure is to identify the users who have similar interests in the tourism domain with the active user. Hence, approximating similarity between the active user and his/her selected friends is performed. The friends who share the similar preferences with the active user are called neighbors. If the active user and neighbors have checked-in the same attractions, the value of similarity between them will be high. To estimate similarity between two users, two similarity measures named Jaccard (1912) and Cosine (Salton and McGill, 1987) are selected. Because most of users check-in a very small number of the entire attractions, the user-attraction matrix is commonly very sparse (Williamson and Ghahramani, 2008). When data is sparse, Jaccard and

Cosine have been suggested to deal with this problem because both of them are able to ignore 0-0 pairs of check-in between the active user and his/her neighbors (Ertöz, Steinbach and Kumar, 2002). Furthermore, Jaccard and Cosine similarity have capability to compute with the binary values in the user-attraction matrix. Ignoring 0-0 pairs of check-in and the binary values are not capable to be applied with Pearson correlation because the average rating scores of the active user and neighbors are adopted in this similarity computation as illustrated in the Equation 2.18. The following example demonstrates the way to estimate the similarity between the active user and a neighbor based on Jaccard and Cosine similarity measures.

		Attractions									
		1	2	3	4	5	6	7	8	9	10
Users	1	1	1	0	0	1	0	0	0	0	0
	2	1	0	1	1	1	1	0	0	0	0
	3	0	0	0	0	1	1	1	1	1	1

Figure 3.15 An example of user-attraction matrix for similarity estimation.

As shown in Figure 3.15, the user-attraction matrix consists of three users and ten attractions. In this example, the first user is assumed as the active user and the others are neighbors. Besides, the example solely explains similarity calculation between check-in histories of the active user and the second user relied on Jaccard and Cosine similarity measures. Jaccard similarity determines the sets of attractions checked-in by the active and the second users for its computation as defined in the Equation 3.10.

$$jaccard(x, y) = \frac{|x \cap y|}{|x \cup y|} \tag{3.10}$$

where x and y are the sets of attractions checked-in by the active user x and the user y and $jaccard(x, y)$ is the result of the two sets' intersection divided by their union.

Jaccard similarity omits 0-0 pairs by only considering a set of attractions checked-in by the two users. For example {*Attractions*$_1$, *Attractions*$_2$, *Attractions*$_5$} is a set of checked-in attractions of the active user and {*Attractions*$_1$, *Attractions*$_3$, *Attractions*$_4$, *Attractions*$_5$} is a set of checked-in attractions of the second user.

With the example, the two sets are

$$x = \{Attraction\,s_1, Attraction\,s_2, Attraction\,s_5\}$$

$$y = \begin{Bmatrix} Attraction\,s_1, Attraction\,s_3, Attraction\,s_4, \\ Attraction\,s_5 \end{Bmatrix}$$

Then

$$|x \cap y| = \{Attraction\,s_1, Attraction\,s_5\} = 2$$

$$|x \cup y| = \begin{Bmatrix} Attraction\,s_1, Attraction\,s_2, Attraction\,s_3, \\ Attraction\,s_4, Attraction\,s_5, Attraction\,s_6 \end{Bmatrix} = 6$$

Hence, the Jaccard similarity is

$$jaccard(x, y) = \frac{2}{6} = 0.33$$

The value 0.33 is a similarity value between the active user and the second user measured by using Jaccard similarity. The range of Jaccard similarity is between 0 and 1, where the value 0 indicates two users are not similar and the value 1 indicates perfect similarity.

In case of the Cosine similarity, it is able to be calculated by using the Equation 3.8. The check-in histories of the active and the second user need to be converted into vectors. The vector conversion can be displayed as follows:

$$x = (1,1,0,0,1,0,0,0,0,0)$$

$$y = (1,0,1,1,1,1,0,0,0,0)$$

As mentioned before in the beginning of similarity estimation, the user-attraction matrix is very sparse and Cosine similarity is able to neglect 0-0 matches of checking-in. Therefore, the high dimensional vectors could be reduced. In this study, the dimensionality reduction starts with merging the checked-in attractions of the two users. For instance, {$Attraction_1$, $Attraction_2$, $Attraction_3$, $Attraction_4$, $Attraction_5$, $Attraction_6$} is a set of integration of checked-in attractions between the active user and the second user. Eventually, the set is exploited to generate the new vectors of the two users as follows:

$$x = (1,1,0,0,1,0)$$

$$y = (1,0,1,1,1,1)$$

Then

$$\|x\| = \sqrt{1^2 + 1^2 + 0^2 + 0^2 + 1^2 + 0^2} = \sqrt{3} = 1.732$$

$$\|y\| = \sqrt{1^2 + 0^2 + 1^2 + 1^2 + 1^2 + 1^2} = \sqrt{5} = 2.236$$

The result of dot product is

$$x \cdot y = (1 \times 1) + (1 \times 0) + (0 \times 1) + (0 \times 1) + (1 \times 1) + (0 \times 1) = 2$$

Hence, the Cosine similarity is

$$cosine(x, y) = \frac{2}{1.732 \times 2.236} = \frac{2}{3.872} = 0.51$$

The value 0.51 is a similarity value between the active user and the second user based on Cosine similarity. The similarity value will range from 0 to 1 where 1 indicates perfect similarity and 0 indicates the two users are not similar.

With the advantage of Jaccard and Cosine similarity as useful measures for sparse data and binary data, this study needs to investigate their performances for personalized attraction recommendation.

In the second procedure of Personalized Attraction Recommendation, check-in histories of all friends selected by three data selections are conducted to compute similarity with the active user. When the second procedure finished Jaccard or Cosine similarity computation, it has a role in selecting the most similar users. The value 15 is the number of most similar users (*k*-nearest neighbors) selected to compute recommendation in the following

procedure. The number of nearest neighbors is obtained from exploring neighborhood-based approach of collaborative filters proposed by Gjoka and Soldo (2008).

3.2.3) Recommendation

The final procedure is recommendation. The intention of this procedure is to offer personalized attractions to the active user by processing his/her preferences with check-in histories of 15-nearest neighbors. With similarity estimation, 15-nearest neighbors have their own value of similarity. The similarity value is also called weight. The weight is adopted to calculate a score of recommended attractions for the active user. The score calculation is introduced by Weiss and Indurkhya (2001). Basically, the CF technique tends to recommend attractions where the active user has never checked-in before. It considers the score in order to sort the personalized attraction in descending order. Then the technique selects the top-*N* personalized attractions with the highest score to recommend the active user. An Equation 3.11 shows the way to calculate the score of personalized attractions.

$$score(a, j) = \sum_{i \in N} W_{a,i} C_{i,j} \tag{3.11}$$

where $score(a, j)$ is the score for an active user *a* and an attraction *j*, $W_{a,i}$ is the similarity between the active user *a* and user *i* based on Jaccard or Cosine approach,

$C_{i,j}$ is the check-in value of user i on attraction j and N is a set of the nearest neighbors.

When computing score is done, the advised outcomes are ready to be recommended to the active user. The last but not least, there are two restrictions operated in the next process in order to limit the scope of recommendation.

3.3) Personalized Attraction Restriction

The last process of Personalized Engine is restricting results of recommendation. This process takes the personalized attractions produced from the previous process to execute limitations. There are two restrictions manipulated in this process. The first one is a restriction of attraction categories. Other one is a distance limitation. As mentioned in Me-Location application, the active users are able to select 11 categories of attractions. Besides, the active users can assign distance of radius in order to limit the number of recommended attractions around the current point of them. Hence, parameters of the two limitations are sent from the client application to manipulate in this process.

The process begins with determining categories of attractions selected by the active user. Table 3.2 demonstrates the list of available categories. These are 11 categories taken from QALL-ME ontology (Ou et al., 2008) to group the different types of attractions. Consequently, each of 11 categories has their own keywords. These keywords are adopted to match with the recommended attractions. For example, the keywords' set of "Restaurant" are "Food", "Pizza", "Dinner",

“Steakhouse”, “Noodle” as well as “Gastropub”. If the active user chooses Restaurant as the desired category, a set of those keywords belonged to Restaurant will be taken to map with the categories of recommended attractions. Hence, the process needs to use additional information of advised attraction from the knowledge base in order to operate keyword mapping. A regular Expression which is proposed by Aho, 1990 based on a PHP program is adopted to map between keywords of selected categories and categories of recommended attractions. If the categories of recommended attractions are matched with the keywords at least once, the attractions are selected. Finally, the filtered results will be sent to limit again by considering their geographical positions in specified radius.

After limiting categories of suggested attractions, Personalized Engine continues to process the radius restriction. In this task, Personalized Engine uses latitude and longitude of the active user for radius calculation. Then it takes a value of distance assigned by the active user in order to compute the radius. After that the latitude and longitude of recommended attractions are adopted as parameters for distance computation. The outcome of computation is a distance in kilometers. If the attractions have their positions in the assigned distance, they are considered to be the final results. Eventually, the mobile engine converts the last consequences and their related information, especially scores and categories into JSON format. These results become responses and they are transferred back to display at Content Presentation as shown in Figure 3.6(a).

3.1.3 System Testing and Evaluation

Testing the mobile engine will be performed by fifteen active users as participants who have smart devices with Apple iOS or Google Android. Safari and Chrome are two recommended browsers for the two mobile OSs. The URL of Me-Locations is sent to these active users for entering the application. The active users must be members of Facebook for registering the application. The application registration enables the mobile engine to access and extract interests of the active users in the tourism domain from SNS servers. In this research, Nielsen's approach (Nielsen, 2000) is adopted to obtain the total number of active users. SNS profiles, friends, check-in history of the users and attractions are input data of the mobile engine for evaluation. The input data are stored in the knowledge base and manipulated to test the system operation, especially category prediction and performance of recommending personalized attractions in the aspect of both quality and time usage.

In this research, there are three issues proposed for the mobile engine evaluation including performance of category prediction, correctness of recommendation and response time of suggestion. With these evaluations, the testing data is separated into two groups definitely. The first group is data of attractions comprising titles and categories. Only the first issue uses this set of data for assessment. On the other hand, the second issue and the third issue adopt the second group of data for appraisement. The second group of data contains users' profiles, social relationships and check-in histories. In case of evaluation metrics, recall is raised in the first assessment to measure the performance of categorization models, particularly the correctness of prediction. The second evaluation measures

the quality of recommendation by using the recall as well. The last one is evaluating the response time for recommendation. Full detail of three evaluations will be described in Chapter 4.

3.2 Research Instruments

In this section, tools for system development are demonstrated as follows:

3.2.1 System Development Instruments

1) Hardware specification includes:

- Processor: Intel Core i5 2410M 2.30 GHz
- Memory: 4 GB
- Hard Drive: 500 GB
- Internet Connection: 802.11g wireless LAN

2) Software specification includes:

- Operating System: Microsoft Windows 7 32bits
- Web Browser: Google Chrome Version 23.0.1271.95 m
- Web Server: Apache Web Server Version 2.2.8
- Programming Language: PHP Script Language Version 5.2.6, Python Version 2.7, NumPy Library Version 1.7.1 and SciPy Library Version 0.12.0
- Database Management System: MySQL Database Version 5.0.51b with phpMyAdmin Database Manager Version 2.10.3
- Application Programming Language: HTML5, jQuery Version 1.10.2 and BootStrap Version 3.0

- Application Development Tools: Sublime Text 2.0

3.2.2 Evaluation Instruments

1) System specification of Apple iOS mobile phone consists of:

- Mobile Brand: Apple
- Model: iPhone 5
- Operating System: iOS 7.1
- Processor: Dual-core 1.3 GHz Swift (ARM v7-based)
- RAM: 1 Gb
- ROM: 16 Gb
- Screen Resolution: 640 x 1136 pixels
- Internet Connection: 802.11g wireless LAN and 3G

2) System specification of Google Android mobile phone consists of:

- Mobile Brand: Sony
- Model: Xperia SP
- Operating System: Android OS Version 4.3
- Processor: Qualcomm® Snapdragon S4 1.7 GHz
- RAM: 1 Gb
- ROM: 8 Gb
- Screen Resolution: 720 x 1280 pixels
- Internet Connection: 802.11g wireless LAN and 3G

3.3 Data Collection

In this work, there are two groups of data testing with the three evaluations. The first group is information of attractions consisting of their titles and categories. The related data of the first group is retrieved from an online system named SMARP. These data is stored in a local database and used for testing correctness of category prediction. The second group employs three data for assessment. The three data are users' profiles, friends and check-in histories. These data are obtained from the fifteen active users. The data acquisition starts when the users allow the mobile engine to access their information on SNS servers via entering Me-Locations application. Then the users are required to update SNS information through the application. After that the mobile engine receives a permission to extract the essential information from SNS servers. Lastly, the extracted data are stored in the local database and all of them will be conducted to evaluate both quality of recommendation and response time of the mobile engine.

3.4 Data Analysis

Data analysis in this research comprises correctness of category prediction, correctness of recommendation and response time analysis.

3.4.1 Analyzing Correctness of Category Prediction

The first analysis is conducted by considering the correctness of category prediction. The attractions with missing categories are revised by labeling the predicted categories. The revising process is performed in the Category Categorization sub-module. The latent semantic analysis (LSA) and machine learning

(ML) are two techniques which are adopted to construct categorization models. The categorization models have duty to forecast categories for the incomplete attractions based on their titles. In this analysis, there are four ML algorithms taken to build the categorization models. The four algorithms are NB, J48, BPNN and SVM. To find the most efficiency approach, the four algorithms need to be compared their performance with each other. In addition, there is a comparison between the different sizes of semantic features obtained from the LSA technique. The *Recall* from information retrieval (IR) science is taken to measure the competency of each categorization model depending on four different algorithms and various volumes of features.

The confusion matrix as shown in Table 3.5 has a role in explaining the evaluation of categorization models. In the matrix, there are four groups of data. These data consists of True Positive (*TP*), False Positive (*FP*), True Negative (*TN*) as well as False Negative (*FN*). *TP* represents the total number of relevant attractions categorized for a particular category. *FP* expresses the total number of non-relevant attractions categorized for a particular category. *TN* indicates the number of non-relevant attractions not categorized for a particular category. *FN* corresponds to the total number of relevant attractions not categorized for a particular category.

Ultimately, the four groups of data are managed to compute recall as displayed in the Equation 3.12 (Sokolova and Lapalme, 2009).

Table 3.5 Confusion matrix represented competency evaluation of categorization models based on IR.

	Relevant Attractions	**Irrelevant Attractions**
Categorized	*TP*	*FP*
Not Categorized	*FN*	*TN*

$$Recall = \frac{TP}{TP + FN} \tag{3.12}$$

3.4.2 Analyzing Correctness of Recommendation

The second analysis is evaluating correctness of recommendation. The correctness is used to indicate the quality of recommendation which the active user will obtain. This analysis takes the recall to achieve the assessment. Recall in this analysis is capable to apply the Equation 3.12 for calculation. Hence, both confusion matrix and four groups of data (i.e., *TP*, *FP*, *FN* and *TN*) introduced in previous analysis are related with this examination. Nevertheless, the confusion matrix displayed in the below table shows the different aspect of implementation. *TP* represents the number of recommended attractions checked-in by the active user. *FP* represents the number of recommended attractions which the active user has never checked-in. *FN* indicates the number of attractions which are checked-in by the active user but not recommended by the system. *TN* expresses the number of attractions which have never checked-in by the active user and not recommended by the system.

Table 3.6 Confusion matrix represented effectiveness and efficiency evaluation of recommendation system based on IR.

	Checked-in	**Not Check-in**
Recommend	*TP*	*FP*
Not Recommend	*FN*	*TN*

As mentioned in subsection 3.2.3, the CF-based recommendation system commonly provides top-*N* personalized attractions which the active user has never checked-in. However, there is not selecting top-*N* recommendation in this evaluation. The evaluation takes all suggested attractions both checked-in by the active user and unchecked-in in order to measure the quality of the recommendation. Furthermore, the appraisement also investigates the impact of three data selections to the performance of the mobile engine. The three data selections consist of finding friends from the entire data, adopting solely SNS friends, and implementing the DF approach to select friends who share the similar SNS profile. The details of performance comparison with different data selections are demonstrated in Chapter 4.

3.4.3 Analyzing Response Time

The last but not the least analysis is the response time analysis. The analysis determines the processing time of the mobile engine in order to measure its performance. Therefore, response time of the fifteen expert users gotten from using Me-Locations application is averaged to display as the overall performance. The response time starts measuring when the active users touch the Recommend Me

button in the home screen of Me-Locations until they are able to see the results in the screen of Content Presentation as illustrated in Figure 3.6(a).

CHAPTER 4

THE RESULTS OF THE STUDY AND DISCUSSIONS

In this chapter, the results of designing a mobile engine for personalized tourist attraction recommendation using SNSs are proposed. As mentioned before in Chapter 3, there are investigations of three data analysis including performance of attraction category prediction, the correctness of attraction recommendation and response time of advice. Hence, the explanation of this chapter is organized according to the three data analysis, respectively. The detail of each data analysis comprises experimental environment, evaluation results as well as summary and discussion. Firstly, the experimental settings focus on a description of dataset, software specification and evaluation methodology. Hardware specification is not mentioned because all of data analysis is conducted with the same machine as denoted in Chapter 3. Secondly, the experimental results reveal the performance of the mobile engine in each analysis. Thirdly, the last part of each examination is summary and discussion. Lastly, this chapter discusses the results of the hypothesis testing.

4.1 The Evaluation of Performance of Category Prediction

The first data analysis has purpose to assess the performance of the mobile engine in the aspect of forecasting the missing category of attractions fetched from SNSs. In this study, a latent semantic analysis (LSA) is a technique

adopted to transform term vectors into latent semantic vectors. Besides, the technique is able to reduce dimensionality before sending the outcome to process with machine learning (ML). Machine learning is a technique used to construct a classification model in order to predict category. There are four algorithms of ML taken to build the classification models. Hence, the evaluation concentrates on the comparison between the four algorithms with different sizes of dimensions. The purpose is to find the most suitable approach for the mobile engine. The following topics are settings of experimental environment, evaluation results as well as summary and discussion. The first one describes a detail of dataset, software specification and assessment method. The second one exhibits experimental results. The results comprise correctness of prediction and model construction time. The correctness of forecast is measured by using recall. Using recall is capable to indicate the capability of models after performing categorization. The construction time displays time usage of each ML algorithm for building the models. The last topic of this section provides summary and discussion.

4.1.1 Experimental Environment

1) Dataset

The dataset conducted in this evaluation is fetched from the SMARP system. It contains 10,250 records of attractions with eleven categories. The eleven categories are gotten from QALL-ME ontology (Ou et al., 2008). It has a role to group the tremendous categories of the 10,250 attractions. Table 4.1 demonstrates the statistics of dataset.

Table 4.1 Statistics of dataset conducted in the evaluation of performance of categorization models.

Category of attractions	Number of attractions
1. Accommodation	1,380
2. Attraction	1,155
3. Coffee & bakery	1,046
4. Convention exhibition	50
5. Entertainment	308
6. Health	255
7. Public	100
8. Restaurant	4,161
9. Shopping	983
10. Sports	250
11. Terminal	562

All titles of attractions are written in English language and most of locations are located in Thailand. Initially, the dataset needs to be transformed into latent semantic space by using LSA technique. Then there is dimensionality reduction performed by selecting top-*k* dimensions of latent semantic space. The purpose of dimensionality reduction is to remove the noise from the semantic space. Furthermore, it is able to decrease computational cost of categorization model construction. However, selecting various sizes of dimensions of latent semantic space should be examined to discover the most suitable number of dimensions. After that, the reduced sets of latent semantic space based on

different sizes of dimensions are divided into training and test set. ML exploits the training set to create categorization models with four different algorithms. The four algorithms include Naïve Bayes (NB), Decision Tree (J48), Back-Propagation Neural Networks (BPNN) and Support Vector Machine (SVM). Finally, the categorization models are manipulated to evaluate quality of prediction with the test set.

2) Software Specification

There are three main packages of software employed in the first evaluation. The first package consists of PHP version 5.4.7, Apache version 2.4.3 and MySQL version 5.5.27 used as a DBMS. This package is implemented to retrieve data of attractions from an online system named SMARP and then store them to a local database. After that, it converts those data into a term-document matrix before performing the LSA. The second package has a duty to execute the LSA algorithm, especially matrix multiplication and SVD execution. It includes Python version 2.7, NumPy library version 1.7.1 and SciPy library version 0.12.0. Semantic vectors as the output of SVD are used to train and test categorization models with the last package of software called Weka version 3.6.10.

3) Evaluation Methodology

The competency of categorization models is evaluated in both aspects including correctness and building time. The correctness of prediction is evaluated by considering a standard measure, called Recall. The standard 10-fold

cross-validation is implemented to train and test those models. All of machine learning algorithms, except Back-Propagation Neural Networks (BPNN), are performed by using the default parameters of Weka. In case of BPNN, there are four parameters consisting of the number of hidden layers, learning rate, momentum as well as epoch. The four parameters are adjusted to 30, 0.1, 0.5, and 50, respectively. On the other hand, the model construction time is evaluated by determining from the time usage of each algorithm based on different volume of dimensions of latent semantic space.

4.1.2 Experimental Results

1) Model Correctness

The correctness of categorization models is compared by selecting various sizes of dimensions *k* from 200, 400, 600, 800, 1000, 1200 to 1400. As illustrated in Figure 4.1, SVM and BPNN are two outstanding algorithms but the strongest one is SVM. The two algorithms get benefit from increasing the number of dimensions from 200 to 1200. Nevertheless with 1400 dimensions, efficiency of these two algorithms is lightly declined. Therefore, the best accuracy of SVM and BPNN with 1200 dimensions is 77.82% and 75.96%. In case of J48, increasing the amount of dimensions led models' performance to be slightly decreased. The highest performance of J48 with 200 dimensions is 67.95%. The most impractical algorithm in this work is NB. With 400 dimensions, the capability of NB algorithm is highest where the algorithm could be achieved in 47.85% of recall. However, with more than 400 dimensions, the proficiency of NB is reduced increasingly.

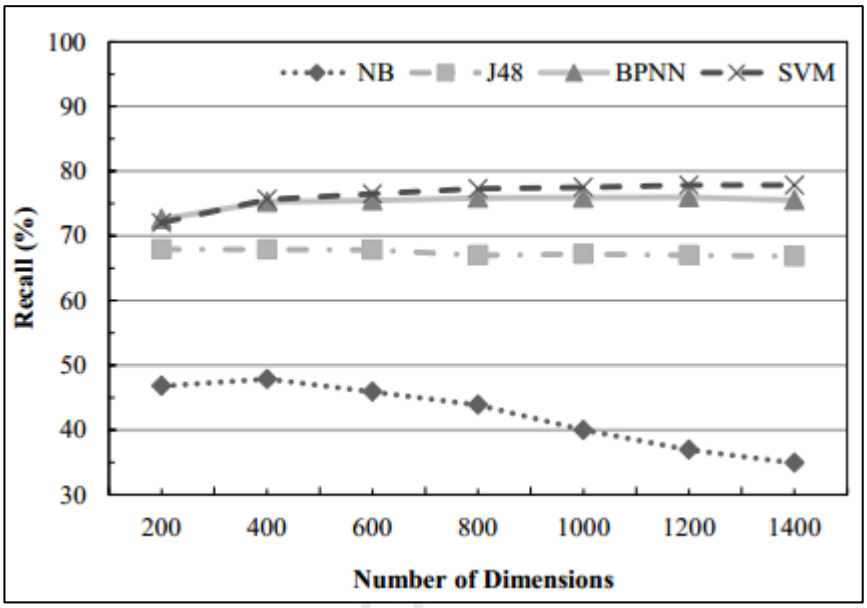


Figure 4.1 The correctness of categorization models compared with various numbers of dimensions and four machine learning algorithms.

2) Model Construction Time

The comparison of model construction time is based on various numbers of dimensions as similar as conducted in the previous evaluation. As shown in Figure 4.2, increasing the number of dimensions makes every ML algorithm to take longer time to acquire categorization models. The NB algorithm takes the shortest time to finish model construction when compared with others. However, the correctness of this algorithm as mentioned above is the most ineffective. In each size of dimensions, SVM and J48 are two algorithms that take approximately equal time to build the models. Nevertheless, the prediction correctness of SVM is better than J48. BPNN is an algorithm which spends the longest time to complete model generation. Even though, the

efficiency of BPNN and SVM is slightly different, the model construction time based on SVM algorithm is less than BPNN.

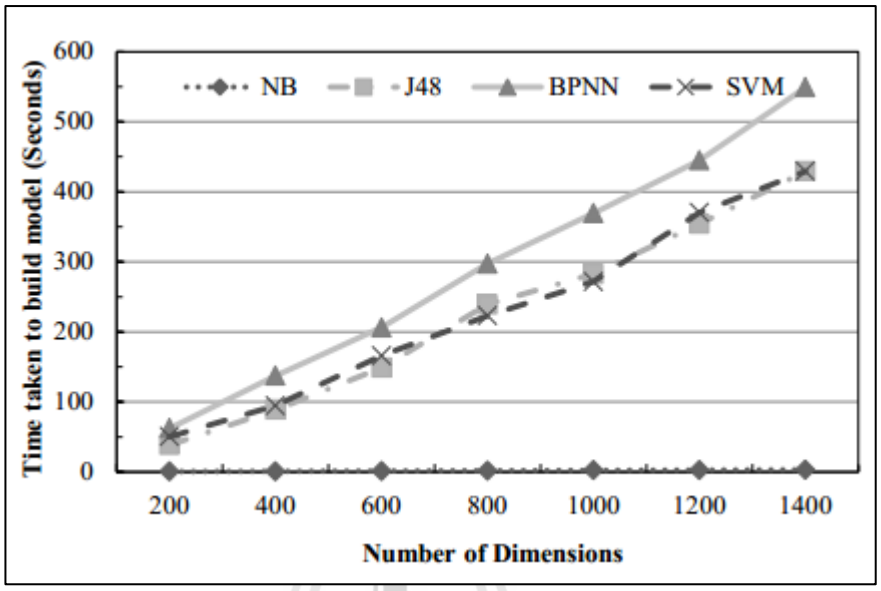


Figure 4.2 The time of model construction compared with various numbers of dimensions and four machine learning algorithms.

4.1.3 Summary and Discussion

To handle with missing or incomplete category of attractions fetched from SNSs, ML plays an important role in dealing with this problem. In this evaluation, 10,250 attractions are retrieved from the SMARP system and regrouped with 11 categories obtained from QALL-ME ontology. An LSA technique is adopted to project a term-document matrix onto latent semantic space by using SVD. The outcome from performing LSA is taken to construct the categorization models for classifying non-labeled attractions. In order to seek the most efficiency approach, there is a comparison of model performance with

different numbers of dimensions and four ML algorithms including NB, J48, BPNN as well as SVM. The experimental results reveal that both SVM and BPNN with 1,200 dimensions of latent sematic space are the outstanding models. They are able to provide the model correctness with 77.82% and 75.96% of recall. Nonetheless, the SVM spends less time than the BPNN to accomplish the model construction.

However in the future, these techniques should be tested with attractions written in non-English language. Besides, the dynamic grouping and classifying categories of attractions should be investigated because there are new kinds of attractions emerged on SNSs every time. This helps the users to discover and select new kinds of attractions conveniently.

4.2 The Evaluation of Correctness of Recommendation

The second data analysis attempts to examine correctness of recommendation. The correctness is measured to indicate the efficiency of the mobile engine in the aspect of recommending personalized attractions to active users. This evaluation adopts three different data selections to perform the recommendation. Detail of datasets is presented in the experimental environment. With distinct size of datasets, the recommendation results are different. This section begins with comparing quality of recommendation based on the three picking data. After that, there is a further examination proposed as the forth data selection. The evaluation results of each data selection have two values based on two similarity techniques named Jaccard and Cosine. Processing similarity is an important procedure of the recommendation. This process has duty to identify *k*-

nearest neighbors for the active users before performing recommendation. In this study, the number of nearest neighbors is 15. The number of nearest neighbors obtained from exploring neighborhood-based CF proposed by Gjoka and Soldo (2008). The consequences of comparison are exploited to identify the most suitable approach for the mobile engine. In case of indicator, Recall is taken to measure the correctness. The second evaluation does not only analyze personalized recommendation but also compare it with non-personalized one. The non-personalized recommendation is offering popular attractions to active users. This technique exclusively considers the popularity of attractions ignoring individual preference of active users. The attractions which have higher frequency of check-ins could be implied that they are more well-known. Lastly, all outcomes of evaluation are presented as the overall performance.

The following topics are the experimental environment, experimental results as well as summary and discussion. The first one introduces detail of datasets, software specification and evaluation methodology. The second one exposes evaluation results based on various approaches as mentioned above. The last one demonstrates summary and discussion of the second data analysis.

4.2.1 Experimental Environment

1) Dataset

Check-in history of users is the dataset retrieved from the SMARP system. It is conducted to process recommendation and evaluation. The dataset mainly has three information comprising users, check-ins and attractions. The first one contains profiles of users and their social relationships, especially

friends. Another one stores data of interaction between users and attractions thus it is able to identify the visited locations of each user. Binary rating is employed to represent the interaction of checking-in, where 1 represents the specific user who has been checked-in the specific attraction and otherwise is represented by 0. The other one keeps the detail of attractions such as name, latitude, longitude as well as category. In this evaluation, the users who have been checked-in greater than or equal to five attractions are considered (Yuan et al., 2013). Table 4.2 illustrates the number of users, attractions as well as check-ins of this dataset.

Table 4.2 Statistics of the entire dataset of the second data analysis.

Dataset	Statistics
Number of users	4,538
Number of attractions	86,237
Number of check-ins	195,824

As mentioned in Chapter 3, testing the mobile engine is performed with fifteen expert users as participants. The experiment considers these users as active users because all users cannot be the active users. The active user in this study is a user who has been registered in the SMARP system and granted permissions to the system in order to access his or her information on SNS. Therefore, the system can identify who are friends of the active users and check-in history of those friends.

In some case of data selection, it uses only SNS friends of the active users to execute recommendation. This means that with the approach, the

mobile engine cannot take unregistered users to achieve suggestion because it cannot know who their SNS friends are. Consequently, 15 active users are proposed to measure overall correctness of recommendation. With difference of data selection, each user has distinct number of input and output data. As described earlier, the forth data selection is a further investigation. Hence, the following table exhibits statistics of each dataset chosen by the first three data selections.

The first data selection exploits entire dataset to perform CF-based recommendation. Using the whole dataset makes all active users to have equal number of friends. Friends in table 4.3 are other users in the SMARP system who could be either SNS friends of the active user or not. In table 4.3, each data selection has different sizes of data, particularly the number of friends. The difference leads volume of attractions checked-in by friends to be dissimilar.

Table 4.3 Statistics of datasets selected by the three data selections.

Active user	Number of attractions checked-in by the active user	First data selection		Second data selection		Third data selection	
		Number of friends	Number of attractions checked-in by friends	Number of friends	Number of attractions checked-in by friends	Number of friends	Number of attractions checked-in by friends
1	11	4,537	195,813	43	1,333	3,466	149,310
2	59	4,537	195,765	227	12,589	3,498	148,382
3	25	4,537	195,799	160	8,872	3,645	162,329
4	12	4,537	195,812	273	10,184	3,720	164,850

Table 4.3 Statistics of datasets selected by the three data selections. (Continued)

Active user	Number of attractions checked-in by the active user	First data selection		Second data selection		Third data selection	
		Number of friends	Number of attractions checked-in by friends	Number of friends	Number of attractions checked-in by friends	Number of friends	Number of attractions checked-in by friends
5	29	4,537	195,795	107	4,566	2,795	114,041
6	44	4,537	195,780	199	9,268	3,590	157,123
7	28	4,537	195,796	318	12,958	3,478	151,355
8	69	4,537	195,755	251	10,756	2,765	111,084
9	15	4,537	195,809	192	8,544	2,388	96,838
10	52	4,537	195,772	345	14,149	3,158	139,395
11	50	4,537	195,774	148	6,623	3,476	147,475
12	45	4,537	195,779	97	5,015	3,446	150,435
13	76	4,537	195,748	180	9,327	3,476	149,569
14	30	4,537	195,794	27	1,576	3,491	151,908
15	8	4,537	195,816	192	10,166	3,471	147,211
Overall	**36.87**	**4,537**	**195,787.13**	**183.93**	**8,395.07**	**3,324.20**	**142,753.67**

The second data selection has the number of friends less than the first one because it determines solely SNS friends of the active users. The fewer friends directly reflect quantity of attractions. For instance, the active users who have more friends could be implied that they are able to have more volume of attractions checked-in by their friends.

The third data selection is a proposed approach in this research. Due to implementing DF approach in the third approach, the number of friends is reduced when compared with the first one. 3,324 is averaged number of friends picked by using the third data selection. The non-equality of dataset affects the mobile engine to provide contrasting quality of recommendation. Therefore, the experimental results demonstrate outcomes of comparison between the three data selections.

2) Software Specification

The evaluation of correctness of recommendation is managed on a local machine with client-server environment. The entire data is exported from SMARP system on February 7, 2013 and imported to this machine. The main package of software applied in this evaluation includes Apache version 2.4.3, MySQL version 5.5.27 as well as PHP version 5.4.7. Apache software facilitates the machine be a web server in order to provide web services. MySQL acts as DBMS for storing and retrieving data. PHP is implemented to conduct both recommendation and evaluation.

3) Evaluation Methodology

An information retrieval measure named Recall is taken to measure competency of the mobile engine for recommendation. The significance of this measurement is to examine how many actually attractions checked-in by the active user could be discovered by the presented recommendation approaches. Recommended attractions are output of suggestion. It is used to

compute recall by measuring with a set of active user's attractions. Thus, recall rate only considers the number of checked-in attractions of the active user appeared in recommendation. Distinction of data selection and similarity calculation are factors which enable recall of each active user to be divergent. The higher recall rate indicates that the mobile engine is able to provide better recommendation to the active user. In this assessment, the Equation of recall is defined in 4.1.

$$Recall = \frac{Number\ of\ attractions\ the\ active\ user\ check\text{-}ins\ in\ recommendation}{Total\ number\ of\ attractions\ the\ active\ user\ check\text{-}ins} \quad (4.1)$$

4.2.2 Experimental Results

This topic introduces experiment results. It starts with presenting the results of personalized and non-personalized recommendation. The outcomes not only display recall of all active users but also expose quantity of recommended attractions. Initially, recall rates obtained from personalized and non-personalized recommendation are illustrated. These recall rates are also separated by following Jaccard and Cosine similarity measures. Table 4.4 displays the example of results. In order to measure recall of non-personalized recommendation, the experiment uses the number of personalized attractions to select top-*N* popular attractions. For example, if the first active user receives 245 personalized locations then the experiment takes this number to pick popular locations for the non-personalized recommendation. Therefore, 245 highest score attractions are chosen to measure recall. After analyzing results from the first three data selection, there is an additional investigation. The investigation picks

top-200 DF-based friends to operate recommendation. Finally, comparison of overall performance is introduced.

1) Comparison of Personalized and Non-personalized Recommendation with Three Data Selections

Table 4.4 depicts the evaluation results using the first data selection. Recall rates as results are separated based on two suggestions (i.e., personalized and non-personalized recommendation) and two similarity approaches (i.e., Jaccard and Cosine similarity). Table 4.4 also presents the number of recommended attractions. With different 15-nearest neighbors, each active user gains advised attractions with different volume. As described above, number of individual locations is exploited to select top-*K* popular locations for measuring popularity-based recall. The results show that the personalized approach provides better quality of suggestion than the other one. This infers that based on popularity, the individual preference of the active user is neglected.

Table 4.4 The results of recommending evaluation performed with the first data selection using all users in the dataset as friends of the active user.

Active User	Number of Recommended Attractions		Personalized Recommendation		Non-Personalized Recommendation	
	Jaccard	Cosine	Jaccard	Cosine	Jaccard	Cosine
1	245	245	90.91	90.91	90.91	90.91
2	660	507	71.19	67.80	33.90	28.81

Table 4.4 The results of recommending evaluation performed with the first data selection using all users in the dataset as friends of the active user. (Continued)

Active User	Number of Recommended Attractions		Personalized Recommendation		Non-Personalized Recommendation	
	Jaccard	**Cosine**	**Jaccard**	**Cosine**	**Jaccard**	**Cosine**
3	348	504	88.00	88.00	28.00	32.00
4	168	209	83.33	83.33	66.67	66.67
5	357	315	55.17	55.17	31.03	27.59
6	575	494	61.36	61.36	20.45	18.18
7	236	214	67.86	67.86	10.71	10.71
8	795	609	68.12	68.12	26.09	21.74
9	296	254	53.33	53.33	26.67	26.67
10	279	212	32.69	30.77	25.00	19.23
11	727	597	62.00	56.00	58.00	56.00
12	442	306	60.00	55.56	24.44	24.44
13	843	508	59.21	53.95	34.21	27.63
14	507	581	76.67	80.00	36.67	36.67
15	210	210	37.50	37.50	12.50	12.50
Overall	**415**	**384.33**	**64.49**	**63.31**	**35.02**	**33.32**

Nonetheless, some users such as the first user deserve equal percentage of recall from the both suggestions. In case of similarity approach, Jaccard similarity has a slightly more capable than Cosine. In case of the first

data selection, the averaged recall of both recommendations based on Jaccard and Cosine similarity are 64.49%, 63.31%, 35.02% and 33.32%, respectively.

In order to examine the effect of determining only SNS friends for recommendation, Table 4.5 exhibits the evaluation results. There are several interesting issues appeared in these results. Firstly, this table has the number of recommended attractions larger than the previous table. This incident leads to answer the problem why popularity-based recall is increased when compared with the same values in Table 4.4. The larger volume of suggested places increases a chance of discovering attractions checked-in by the active users in suggested results. Secondly, many active users do not receive the impact of removing non-SNS friends such as the first, the second and the ninth users. Notice that considering only SNS friends does not make the overall recall to be obviously different from using the entire dataset. This indicates that the social relationship has a significant influence on checking-in of some active users. Lastly with individual suggestion, Jaccard similarity still presents higher recall than Cosine similarity. On the contrary, Cosine similarity is able to deliver better performance than Jaccard similarity for popularity-based recommendation. 63.68% and 63.42% of recall could be obtained from the personalized approach based on Jaccard and Cosine similarity, respectively. The non-personalized approach offers 36.55% and 36.58% of recall.

Table 4.5 The results of recommending evaluation performed with the second data selection using only SNS friends as friends of the active user.

Active User	Number of Recommended Attractions		Personalized Recommendation		Non-Personalized Recommendation	
	Jaccard	**Cosine**	**Jaccard**	**Cosine**	**Jaccard**	**Cosine**
1	307	307	90.91	90.91	90.91	90.91
2	976	938	71.19	71.19	37.29	37.29
3	465	487	84.00	84.00	32.00	32.00
4	210	296	75.00	75.00	66.67	75.00
5	422	384	65.52	65.52	31.03	31.03
6	588	521	61.36	61.36	20.45	20.45
7	384	384	57.14	57.14	21.43	17.86
8	649	609	66.67	66.67	26.09	21.74
9	521	521	60.00	60.00	26.67	26.67
10	385	280	44.23	40.38	25.00	25.00
11	551	529	46.00	46.00	54.00	54.00
12	608	608	60.00	60.00	26.67	26.67
13	786	756	47.37	47.37	34.21	34.21
14	1036	1036	63.33	63.33	43.33	43.33
15	617	617	62.50	62.50	12.50	12.50
Overall	**577.36**	**551.53**	**63.68**	**63.42**	**36.55**	**36.58**

The third data selection integrates the DF approach to select friends who share similar profile with the active users. The approach employs SNS profile of the active users to compute the similarity with others by using

Cosine similarity. Gender, relationship status, living location as well as educational information are instances of profile information. The value of DF similarity is between 0 and 1 where a value approaching 1 means a strong likeness. After computing the similarity, friends whose similarity values greater than 0 are taken to perform recommendation.

Table 4.6 demonstrates the evaluation results. Several interesting issues appear in the results. Although the number of recommended attractions is not quite different from the same values in Table 4.4, the recall of non-individual advice are clearly contrast. Unfortunately, using the DF approach lets recall of some active users to be decreased, especially the first user. This incident makes the total popularity-based recall to be lower than same values in the two previous results. Nonetheless, when compared with the second results, many users get the benefits from adopting the DF technique such as the forth, the eleventh and the thirteenth users. This infers that checking-in of some active users does not depend on the domination of SNS friends. Even though, the performance of non-individual way in this evaluation is not better than the two prior results, the quality of personalized suggestion is nearly equal. Eventually, Jaccard similarity offers finer performance than Cosine similarity. 64.03% and 62.50% are recall of personalized suggestion and another one is achieved in 29.70% and 28.36% of recall.

Table 4.6 The results of recommending evaluation performed with the third data selection using a DF approach to select friends of the active user.

Active User	Number of Recommended Attractions		Personalized Recommendation		Non-Personalized Recommendation	
	Jaccard	**Cosine**	**Jaccard**	**Cosine**	**Jaccard**	**Cosine**
1	232	232	72.73	72.73	9.09	9.09
2	701	377	72.88	67.80	33.90	28.81
3	446	520	84.00	84.00	32.00	36.00
4	173	215	83.33	83.33	66.67	66.67
5	363	373	62.07	58.62	31.03	31.03
6	586	494	63.64	61.36	20.45	18.18
7	227	212	64.29	67.86	10.71	10.71
8	727	558	68.12	68.12	26.09	26.09
9	268	357	53.33	53.33	26.67	26.67
10	258	202	38.46	28.85	23.08	17.31
11	777	852	62.00	60.00	58.00	58.00
12	467	284	62.22	60.00	24.44	20.00
13	822	508	59.21	53.95	34.21	27.63
14	497	581	76.67	80.00	36.67	36.67
15	256	256	37.50	37.50	12.50	12.50
Overall	**453.33**	**401.40**	**64.03**	**62.50**	**29.70**	**28.36**

This evaluation results leave a question what would be happen if this experiment based on the DF approach takes the same quantity of data as shown in the second data selection based on the SNS approach. Hence, the next

topic presents the further examination of the forth data selection to find the answer for this question.

2) Further Investigation with 200 DF-based Friends

The further investigation is introduced as the forth method of data selection. The objective is to observe quality of recommendation gotten from the forth method when DF-based friends are decreased. Because the average number of friends provided by the second data selection is 183.93, this examination selects top-200 based friends to perform the recommendation and evaluation. Table 4.7 depicts statistics of dataset relied on the forth data selection.

As shown in Table 4.7, the number of friends is 200 and the average number of attractions checked-in by friends is 8,204.07. These numbers are nearly equal to the values gotten from selecting data with the second approach as displayed in Table 4.3. When the evaluation is finished, the results are presented in Table 4.8.

Several interesting issues are occurred in the results. The first issue is the comparison between the forth and the second data selection. Even though the forth one has a capability to furnish the total number of advised attractions larger than the second one, it could not guarantee the better performance. Hence, the second data selection gets approximately 23% better performance than the forth one. Reduction of DF-based friends obviously hurts overall recall, particularly in case of the first, the second, and the fifteenth users.

Table 4.7 Statistics of datasets selected by the forth data selection.

Active user	Number of attractions checked-in by the active user	Number of friends	Number of attractions checked-in by friends
1	11	200	8,424
2	59	200	6,839
3	25	200	9,893
4	12	200	8,969
5	29	200	7,430
6	44	200	9,728
7	28	200	8,833
8	69	200	7,201
9	15	200	7,193
10	52	200	8,877
11	50	200	7,365
12	45	200	8,676
13	76	200	8,215
14	30	200	8,330
15	8	200	7,088
Overall	**36.87**	**200**	**8,204.07**

Table 4.8 The results of recommending evaluation performed with the forth data selection using 200 DF-based friends as friends of the active user.

Active User	Number of Recommended Attractions		Personalized Recommendation		Non-Personalized Recommendation	
	Jaccard	**Cosine**	**Jaccard**	**Cosine**	**Jaccard**	**Cosine**
1	829	829	9.09	9.09	9.09	9.09
2	560	511	44.07	42.37	30.51	28.81
3	633	673	80.00	80.00	36.00	40.00
4	362	521	75.00	75.00	75.00	75.00
5	536	536	55.17	55.17	31.03	31.03
6	327	327	31.82	31.82	11.36	11.36
7	381	381	32.14	32.14	21.43	21.43
8	554	442	30.43	28.99	21.74	20.29
9	543	625	26.67	26.67	26.67	26.67
10	670	626	38.46	36.54	30.77	25.00
11	742	725	52.00	52.00	58.00	58.00
12	695	534	44.44	42.22	28.89	24.44
13	1082	1000	30.26	26.32	36.84	36.84
14	588	679	33.33	36.67	36.67	36.67
15	337	337	25.00	25.00	12.50	12.50
Overall	**604.14**	**597.50**	**40.53**	**40.00**	**31.10**	**30.48**

This indicates that checking-in of those users prefers SNS friends to DF-based friends. Unfortunately, filtering the top-200 DF-based friends is able to get rid of some dominating SNS friends of the active users. The next issue is a comparison between personalized and non-personalized recommendation. The

quality of individual way is better than the other way around 9% of recall. The last issue is about similarity approach. Cosine similarity is not capable to provide recall higher than Jaccard similarity. With Jaccard and Cosine similarity, the personalized approach contributes overall recall with 40.53% and 40% and non-personalized approach serves overall recall with 31.1% and 30.48%.

3) Comparison of Overall Correctness

Figure 4.3 illustrates the overall correctness of recommendation in the form of bar chart. It gathers the averaged recall of the four data selections to display. As shown in Figure 4.3, the first three data selections provide an approximately equal percentage of individual-based recall. The efficiency of the forth selection is lower than the first three selection by roughly 23% of recall. With Jaccard similarity, the highest competency is provided by using the first data selection. The secondary is the third one and the following are the second and the forth one. When considering quality of personalized advices using Cosine similarity, the first is adopting the entire dataset. The second is using SNS friends. The third is exploiting DF-based friends. The last is employing top-200 DF-based friends. Total quality of suggestion based on the non-personalized approach is less than the personalized approach. Briefly, non-individual way with Jaccard similarity provides 35.02%, 36.55%, 29.7% and 31.1% of recall. Implementing Cosine similarity with the popularity-based method offers 33.32%, 36.58%, 28.36% and 30.48% of recall.

Observe that with individual recommendation, the first three data selections are able to provide approximate correctness. Nonetheless, the best

quality obtained from the first data selection could not guarantee that it is the best solution for the mobile engine. The further dimension needed to survey is response time. With the difference of data size, the response time plays a key role in considering what the most suitable recommendation approach for the mobile engine is. Consequently, the next section explains the evaluation results of response time.

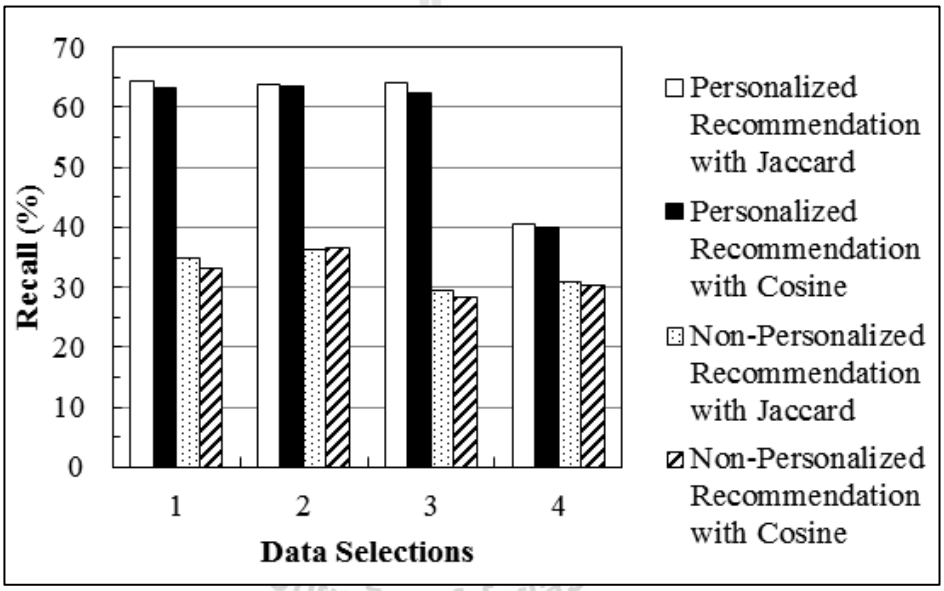


Figure 4.3 Overall correctness of recommendations implemented with the four data selections.

4.2.3 Summary and Discussion

The second data analysis has an intention to evaluate the correctness of recommendation with various data selections. Recall is adopted to measure the correctness. There are four data selections raised to perform suggestions. Besides, each performing suggestion also implements with two similarity

approaches named Jaccard and Cosine. The similarity approach has duty to calculate the likeness between users. The evaluation results reveal that personalized method has more competent than non-personalized method. The difference between them is approximately 23% of recall. Furthermore with personal suggestion, the comparison shows that the first three data selections provide the approximate quality but the forth one does not. The results also indicate that Jaccard similarity has more capable than Consine similarity. However, the highest efficiency is gotten from performing personalized advice with Jaccard similarity by using the first data selection. This incident could not confirm that using the entire dataset is the best solution for suggestion. Hence, the next analysis is examining response time of personalized recommendation with these four data selections.

4.3 The Evaluation of Response Time

As described in the previous evaluation, the first three data selections are able to provide nearly equal quality of suggestion. This cannot lead to decide what the best approach for the mobile engine is. Thus, the last evaluation is performed to find the answer of this question because the different numbers of datasets affect time usage of suggestion. Objective of this assessment is to measure response time of recommendation. The response time is calculated from starting the recommendation until the active users receive the results. This section begins with the experimental environment. Then it presents experimental results and the last presentation is summary and discussion. The experimental settings solely focus on evaluation methodology because the experiment adopts both dataset and

software as similar as the prior evaluation. After that the experimental results demonstrate the response time of recommendation with the four various data selections. Summary and discussion of this evaluation are the last presentation of this section.

4.3.1 Experimental Environment

Both dataset and the set of software from the previous evaluation are taken to manipulate in this appraisement. The four datasets based on the four selections and three softwares (i.e., Apache version 2.4.3, MySQL version 5.5.27 and PHP version 5.4.7) are conducted to perform this experiment. The statistics of datasets picked by the four selections are shown in Table 4.3 and Table 4.7. Therefore, this topic does not explain the both experimental settings in detail. It solely concentrates on evaluation methodology.

In evaluation methodology, the response time is adopted to measure how long the active user waits for recommendation. This evaluation only considers response time of personalized approach because non-personalized method does not take preferences of users to process suggestion. Consequently, the response time has two values based on Jaccard and Cosine similarity measures. The unit of response time is second. All response time of active users is averaged to be the overall time as displayed in Figure 4.4.

4.3.2 Experimental Results

The results of response time comparison are depicted in Figure 4.4. The four data selections demonstrate distinct time usage for recommendation.

With adopting the whole dataset, the mobile engine takes the longest time to achieve recommendation. Identifying 15-nearest neighbors from 4,537 users is the major reason that makes active users to wait around 14 to 17 seconds. Employing only SNS friends has a capability to provide the fastest suggestion. The active users are able to get their personalized attractions within two seconds approximately. Because the average number of SNS friends is 183.93, the mobile engine is capable to provide outcomes with this second data selection. This speed is faster than others. By the way, integrating the DF approach to reduce the volume of friends is capable to enhance the response time. With fewer friends, this third data selection is able to finish recommendation faster than the first one. Moreover, using the 200 DF-based friends reveals that it could aid the mobile engine to take a shorter time in order to complete suggestion than the prior DF approach. Unfortunately, the final data selection hurts quality of recommendation as shown in Figure 4.3. Even though using 200 DF-based friends makes the mobile engine to be faster, its response time is slower than using SNS friends. This incident could be explained that the mobile engine requires computing profile-based similarity between the active user and all users before processing the suggestion. On the contrary adopting solely SNS friends, the mobile engine does not have any pre-processing to perform hence it could operate suggestion instantly.

When comparing the response time of the four data selections, these results indicate that the processing time of recommendation depends on two major factors. The first factor is the number of friends and their checked-in attractions. With the larger number of data in the first factor, the mobile engine spends more time to achieve the recommendation based on the user-based CF

approach. The second factor is the preprocessing time of user-based CF approach. The second factor has obviously exposed when comparing between the second and the forth data selections. The nearly equal number of friends and their checked-in locations between the two data selections does not guarantee the same results of response time. The preprocess of CF approach adopted in the fourth data selection is the DF approach in order to seek the friends who share mutual demographic profile with the active users, whereas the second data is selected for SNS friends. Therefore, the effect of the preprocess of CF approach causes the forth data selection take a longer time than the second one to complete the recommendation.

In case of similarity technique, Cosine implementation requires time to compute similarity more than Jaccard because it has more complex calculation. Hence, the overall response time based on the Cosine similarity is slower than the Jaccard similarity.

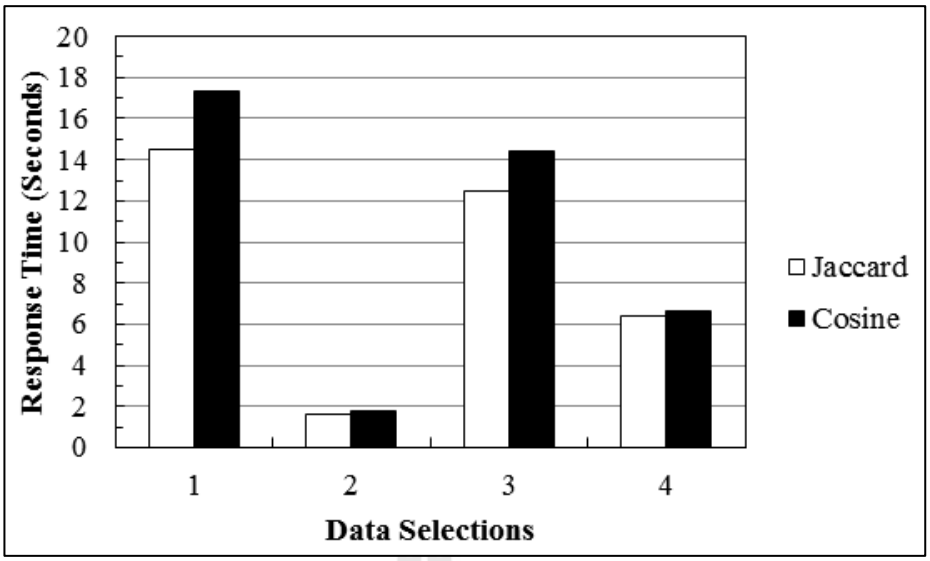


Figure 4.4 Overall response time of recommendations implemented with the four data selections.

As mentioned in the evaluation of correctness of recommendation, the first three data selections are able to provide almost equal quality of suggestion. This situation could not decide what the best approach between the three is. Therefore, the third data analysis is examined to discover the most suitable approach for the mobile engine. When considering both correctness and response time of suggestion, the best approach is adopting the second data selection based on SNS friends. The second is the third data selection based on the DF approach. The last is the first data selection using the entire dataset.

4.3.3 Summary and Discussion

The purpose of the third data analysis is to measure the response time of recommendation. The evaluation results have a benefit to identify the

best solution for the mobile engine. This analysis takes datasets and softwares from the previous examination to manipulate evaluation. The results reveal that exploiting SNS friends enables the mobile engine to be the fastest recommender. Besides, implementing the DF approach is able to improve speed of suggestion. In contrast, using the whole dataset causes the mobile engine take the longest time to perform suggestion. There are two factors affecting the response time of the mobile engine. The first one is the number of friends and their checked-in attractions. The last one is the preprocessing time of the user-based CF approach. Furthermore, the results exhibit the response time of the mobile engine based on Jaccard and Cosine similarity. The complexity of Cosine leads this technique to consume more time than Jaccard. When determining both response time and quality of recommendation, the most appropriate approach is adopting SNS friends to perform the tourist attraction recommendation.

4.4 The Results of the Hypothesis Testing

As mentioned in Chapter 1, this study raises three research hypotheses as follows. Firstly, the mobile engine can revise the incomplete categories of attractions on SNSs by correctly with greater than 80% of recall. Secondly, the mobile engine can provide personalized attractions that match individual travelers' interests correctly with greater than 80% of recall. Finally, the mobile engine is able to response the users by illustrating feedbacks within 5 seconds.

The First Research Hypothesis: the assessment results of the attractions' category prediction exhibit that integrating the latent semantic analysis with

machine learning in order to categorize the missing categories. Nearly 80% of recall gotten from adopting SVM and BPNN algorithms is the best results. Both algorithms with 1,200 dimensions of latent semantic space are the most outstanding models for categorization. 77.82% and 75.96% are the overall recall gotten from these two algorithms. Other two algorithms in this evaluation are J48 and NB. They have capability to provide recall rates with 67.95% and 47.85%.

The accuracy of categorization does not only depend on learning algorithms but also rely on size of input data. As shown in Table 4.1, the number of attractions in each category is not equal. With this case, classifiers might label some locations in “Health” to be “Restaurant”. This means that the higher volume of locations in the restaurant category is capable to dominate some categories with smaller amount of data when it is used to build the categorization models.

With more than 75 percent of recall, the first research hypothesis can be concluded that the results are able to compromise on this hypothesis although it cannot match perfectly. To implement with the mobile engine, the SVM algorithm with 1,200 dimensions is considered to construct the categorization model.

The Second Research Hypothesis: the evaluation results of the second data analysis reveal that 80% of recall as mentioned in this hypothesis is big challenge. Following the results, adopting the first three data selections with Jaccard similarity can offer recall with 64.49%, 63.68% and 64.03%, respectively. This implies that the individual interest of user is the main

challenge of this research. With implementing the CF approach, the personalized attractions of active users are obtained from the nearest-neighbors who share similar check-ins. Due to the different checking-in style of each user, it is very difficult to find other users who share 80% of similarity with each active user. Although the recommendation uses the entire data, the maximum of recall is 64.49%. In addition, there are many facts appeared in this observation. Some active users get benefit from using SNS friends for recommendation. Despite removing the large number of non-SNS friends, these users still obtain high quality of recommendation. On the other hand, some of them gain the advantage from adopting the DF approach. In this case, social friendship does not improve quality of recommendation for these users. Hence, the DF approach can help them to get better recommendation. Unfortunately, adopting top-200 DF approach is capable to hurt recall of some active users and causes the overall recall to be declined.

Following the experimental consequences, the second research hypothesis can be concluded that the consequences do not conform this hypothesis at 80% of recall. Even though the outcome is not consistent with the goal, it is able to demonstrate the possible way to boost recall when compared with the previous endeavors. In this research, using social relationship exposes the most efficiency of recommendation. Nevertheless, checking-in of users does not only depend on their SNS friends but also relies on the personal desire. In order to enhance the correctness, the social relationship and the nature of checking-in need to investigate together.

The Third Research Hypothesis: the evaluation outcomes of the third data analysis reveal that manipulating both SNS friends and DF-based friends can facilitate the mobile engine to decrease response time of recommendation. Time usage of suggestion relies on the number of dataset in the recommendation. The larger number of dataset causes the mobile engine take more time to complete advices. Performing recommendation with the entire dataset definitively takes the longest time to be complete. The intention of this hypothesis is to provide personalized attractions for active users within 5 seconds. The experimental results expose that there is only one approach which can accomplish the suggestion in 5 seconds. The approach is adopting SNS friends. The average time of this approach is 2 seconds. By the way, utilizing top-200 DF-based friends makes response time to almost reach the goal of this hypothesis. However, the recall gotten from this approach is reduced. With using the third data selection, the DF approach is able to prove that it can decrease response time of the mobile engine. With these results, the third research hypothesis can be concluded that these results are inconsistent with the hypothesis except using SNS friends.

CHAPTER 5

CONCLUSIONS AND RESEARCH RECOMMENDATIONS

The final chapter presents a summary of this study. It begins with describing the summary of research findings. Then the limitation of the study is revealed. After that, the chapter explains the application of the study. The last is suggestion for further study.

5.1 Summary of the Research Findings

This research has the objective to design a mobile engine for personalized tourist attraction recommendation using social networking services. It proposes a mobile engine in order to provide attractions based on individuals' interests by adopting SNSs. The users are able to receive individual attractions from the mobile engine through their mobile devices. Initially, the users enter an application named "Me-Locations" via a browser on their mobile devices. Me-Locations as a one part of the mobile engine is adopted to interact between the users and the mobile engine. The application has two main functions comprising updating SNS data and providing personalized recommendation.

With the first usage, the active users need to grant their permissions to the first function. This action enables the mobile engine to access and retrieve users' information from SNSs. The information consists of three data such as profile, social

relationship, and check-in history. The fetched data are conducted to perform recommendation. After the mobile engine fetches SNS data and stores them in its database, it needs to further investigate a category of attractions because some of them are missing. Labeling the incomplete data of locations is a one task of objective in this study. This task initially exploits location-based SNS (LBSNS) to identify the missing category. Then some of locations which cannot be labeled by using LBSNS are taken to operate categorization. The categorization applies a latent semantic analysis (LSA) and machine learning (ML) to forecast a type of locations based on their title written in English language. There is a comparison between various factors to seek the best approach for the categorization. All classified data eventually are reserved in a knowledge base as data storage of the mobile engine. After the active users update their SNS information, the second function of Me-Locations can be operated.

The second function has a role in personally recommending tourist attractions to the active users. This function enables the active users to view these attractions on a digital map based on their current location. This research adopts collaborative filtering-based approach to deliver attractions individually. Other tasks of objective in this research are relied on performance of the second function. In order to improve quality of recommendation, there are many experiments performed to find the most suitable approach, especially data selections and similarity measures. Finally, the mobile engine exploits the most appropriate approach to provide attractions by sending the consequences to display at mobile devices of the active users.

Many endeavors attempt to enhance quality of location recommendation based on LBSNS with various influences. These influences are time awareness, geographical and social impact. However, this study tries to investigate different impacts both social

relationship and mutual demographic profile. In case of demographic profile, it is very interesting to implement because it is one of significant elements of being SNS. Adopting the DF approach for location recommendation could illustrate a possible way to enhance performance of suggestion. Hence, both using SNS friends and DF-based friends are studied together in this research in order to find the difference between them.

To recommend personalized attractions based on current locations of active users, the mobile engine is manipulated to support mobile devices and operate with the client-server environment. It has Me-Locations as a client application in order to interact with the active users. The application is implemented with HTML5, jQuery and Bootstrap framework in order to support presentation on various sizes of mobile screen. In server side, the mobile engine has many modules for operating suggestion. There is a sub-module named category categorization is implemented with Python language. Nonetheless, most of modules apply PHP language for implementation. In order to support a procedure of these modules, Apache Web Server and MySQL are raised. Apache Web Server is a program enabling the local machine to be able to provide web services. MySQL acts as DBMS is used to manage data storage and retrieval of the mobile engine.

In this study, there are three evaluations of the mobile engine including performance of category prediction, correctness of recommendation as well as response time of suggestion. The first one brings recall from information retrieval science to do measurement. Recall is used to measure correctness of category prediction gotten from distinct categorization models. Even though the second one adopts the measure as same as the first one, recall between these two evaluations are different. Recall of the first

one considers how many attractions in test set could be correctly predicted. Recall of the second one is used to measure correctness of recommendation. It considers how many actually attractions checked-in by the active users could be discovered in recommendation. The last appraisement determines the response time to judge performance of suggestion based on distinct data selections.

The research findings provide the summary as follows:

5.1.1 To handle with incomplete categories of attractions fetched from SNS, ML is proposed to predict the categories. It starts with retrieving 10,250 attractions from the SMARP system. Then these attractions are regrouped with 11 categories based on QALL-ME ontology. LSA as a pre-processing technique is adopted to project a term-document matrix onto latent semantic space by using singular value decomposition (SVD). The ML employs the outcome of LSA to construct categorization models. NB, J48, SVM and BPNN are four algorithms adopted in model construction. The quality of models is evaluated by using recall. The evaluation results indicate that both SVM and BPNN algorithms with 1,200 dimensions of latent semantic space are the two most efficiency approaches. They are able to provide the performance with 77.82% and 75.96% of recall. The finest recall rates of other two algorithms are 67.95% and 47.85% of recall with 200 and 400 dimensions, respectively.

5.1.2 The correctness of recommendation is experimented with four different data selections and two similarity measures named Jaccard and Cosine. The first data selection is using the entire dataset. The second one is adopting only SNS friends. Another one is implementing a DF approach to select friends. Other one is applying top-200 DF-based friends. The evaluation results reveal that the first three data

selections have capability to provide the correctness of suggestion higher than the other. With Jaccard similarity implementation, the three data selections deliver 64.49%, 63.68% and 64.03% of recall. In case of Cosine similarity, they offer recall with 63.31%, 63.42%, and 62.50%.

5.1.3 Although the first data selection is able to provide the best correctness of advice, in case of response time it is not the finest approach. The last evaluation results of this research indicate that adopting both SNS friends and DF-based friends enables the mobile engine to complete suggestion more rapid than the first approach. The faster response time is the result of reducing the number of friends with these two approaches. Nevertheless, the most rapid approach for recommendation is using solely SNS friends. With the second approach implementation, the mobile engine is able to provide personalized attractions within 2 seconds.

Overcoming the information overload problem for tourists in Web 2.0 era is the major motivation of this research. Design and evaluation of the mobile engine indicate that the design of the mobile engine is consistency with the proposed motivation. Using the entire data makes the active users to wait a long time for deserving recommendation. Furthermore, applying popularity-based suggestion could provide plenty of results which do not match the personal desire of the users. The occurrence could lead the users to obtain a bad experience on the recommendation system. The mobile engine demonstrates the possible way to provide attractions associated with individual interests of each user. It facilitates the users to avoid undesired results and aids them to discover the new places offered from other users who share similar tastes of travelling. In addition to personalized attraction recommendation, the mobile engine

is able to limit the number of suggested locations by determining a radius around the current point of the active users.

5.2 The Limitation of the Study

This section presents the limitation of design of the mobile engine for personalized tourist attraction recommendation using social networking services. The detail of limitation is able to be explained as follows.

5.2.1 This research has intention to investigate user information on SNSs in order to provide individual attractions. Hence, the mobile engine needs the active users' permission to access and retrieve their information on SNSs. User profiles, social relationships, and check-in history are the information which the mobile engine needs. The application named Me-Location is developed to facilitate the user in order to allow his or her permission to the mobile engine. In case of social relationships, the mobile engine has right to access only one level of the relationships. This means that the mobile engine is not capable to access data belonged to friends of the active user's friends. Therefore, only active users can deserve recommendation with using SNS friends. In this research, solely fifteen participants register the system in order to evaluate the performance of the mobile engine.

5.2.2 In order to obtain the personalized attractions located nearby the active users' current location, the users must share their current spot by activating wireless networks or GPS function on their smart devices. If they do not share, they cannot see any outcomes of suggestion. Initially, the mobile engine gets latitude and longitude of the active users from this operation. Then, these two geographical values are used as the center point for circle radius calculation. After that, the calculation is performed to

select the personalized attractions within the specific radius defined by the active users. Ultimately, the mobile engine transfers the filtered attractions to display at the Me-Locations application.

5.2.3 The client application of the mobile engine named Me-Locations is developed to support mobile devices with Apple iOS and Google Android. These are two major operating systems (OS) available on smartphones. To use the application, the active users are required to open a web browser on their smart devices. They have their own web browser. Apple iOS has Safari Browser and Google Android has Chrome Browser. Testing the mobile engine with those two browsers exhibits the different ways to allow user's current locations. However, it is not a big problem. Lacking of testing with other mobile operating systems such as Window Phone, BlackBerry and webOS could be the limitation of this study.

5.2.4 The mobile engine only supports English language. This limitation affects revising the missing category of attractions retrieved from SNSs because the revision solely considers the title of attractions written in English language. Thus, the mobile engine does not have capability to categorize non-English attractions.

5.3 The Application of the Study

This research demonstrates many benefits to the tourists in Web 2.0 era. In this era, the information overload obviously annoys the tourists when they need some information what they desire. With personalized recommendation, the tourists are able to obtain attractions matched with their personal interests. Furthermore, the suggestion can lead the users to discover the new locations where they do not know it before. The mobile engine does not only provide individual locations but also identifies other

people who share similar lifestyles of travelling with the active users. This makes the users to have chances in order to explore the group of these people. In addition to recommendation, the mobile engine has capability to categorize attractions based on their title. The notion of text categorization is able to implement with other domains of contents. For example, it can be adopted to classify interests of users based on their description, news, movies as well as lifestyles of tourists. Using SNS friends and DF-based friends for recommendation shows the possible ways to enhance the users' experience when they want recommending results. These techniques are able to reduce response time of advice. The concept of the two techniques could be adapted with any recommendation systems based on the relationships of people on the Internet.

5.4 Recommendations for Further Study

The following suggestions exhibit several issues needed to further investigate for improvement.

5.4.1 Supporting non-English language of the mobile engine. Because there are many attractions written in non-English languages stored in the SMARP system, Examples of those languages are Thai, Chinese, Japanese, Korea and German. The mobile engine could have competency to categorize those attractions. This leads to the question "Can the mobile engine still provide good performance of categorization with non-English language attractions?". Therefore, testing text categorization proposed in this study with non-English language attractions is necessary. Furthermore, automatic grouping categories should be an additional ability of the mobile engine because many categories have a similar meaning. Besides, the new kinds of attraction are able to

emerge every time on SNSs. These enhancements could help the tourists to discover and select new kinds of attractions conveniently.

5.4.2 Optimizing SNS data revision. Due to tremendous SNS data of each active user, this study implements multithreaded programing (MP) proposed by (Chatcharaporn, Angskun, and Angskun, 2013) in order to revise the SNS data. Most SNS data is friend data of active users. The friend data consists of a profile and check-in history. With MP implementation, data of friends is retrieved and split into n-set equally before sending them to execute a revision with n-worker pages. The worker pages have a role to conduct those data by using them to fetch the latest information from SNS servers. The latest feedbacks are transferred to store in a database of SMARP system. With equal sets of worker pages, the separation can be optimized because each active user has different sizes of friend data. The optimization is generating the dynamic number of worker pages for each set of data to perform the revision. It could relieve the workload of server, especially in case there are many SNS data revisions performed at the same time.

5.4.3 Analyzing additional profiles of SNS users. DF approach implementation in this study is able to demonstrate the possible way for improving competency of personalized recommendation. However, the study picks several features of profile to operate the DF approach. These features include gender, relationship status, living location as well as educational information. It is very interesting to try investigation with other features of SNS users and see their impact on quality of recommendation. In addition to profile data, other relationships on SNSs between objects and users are attractive as well. The objects might be movies, books, music, TV shows, sports as

well as groups of interests. These objects could be adopted to identify preferences of the users associated with their check-ins.

5.4.4 Adopting the category-based recommendation. Commonly, check-in history of users includes latitude and longitude of attraction as well as check-in timestamp. If the missing category of attractions is not a problem, the category-based suggestion is a very useful service for tourists. The service could be implemented with a time-awareness approach. For instance, people prefer to go dining places in the evening and some of them would like visit to outdoor attractions on the weekend. Category-based and time-awareness implementation enables the mobile engine to be able to offer more exact attractions in specific time. Moreover, it can be used to reduce the number of recommended items to the users because considering both time and category of attractions is capable to limit the scope of recommendation.

5.4.5 Implementing other forms of rating for CF-based recommendation. In this study, binary rating is taken to represent a relationship between a user and an attraction. 1 represents the user has been checked-in the attraction and 0 is otherwise. Furthermore, the relationship is adopted to find both k-nearest neighbors and scores of recommended items. Basically, the CF approach is capable to be adopted with various formats of rating such as frequency of check-ins, scalar rating or binning rating. This issue needs to further investigate because the various forms of rating let the feedbacks of recommendation to be dissimilar.

REFERENCES

Aho, A. V. (1990). Algorithms for finding patterns in strings. In J. van Leeuwen (ed.). **Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity** (pp 255–300). US: MIT Press.

Ardissono, L., Felfernig, A., Friedrich, G., Jannach, D., Schafer, R., and Zanker, M. (2001). Intelligent Interfaces for Distributed Web-Based Product and Service Configuration. In **Proceedings of the 1st Asia-Pacific Conference on Web Intelligence: Research and Development 2001** (pp 184–188). Berlin, Heidelberg, Germany: Springer.

Azak, M. 2010. **A framework to develop knowledge-based recommenders in cross domains**. M.Sc. thesis, Middle East Technical University, Turkish.

Bao, J., Zheng, Y., Wilkie, D., and Mokbel, M. F. (2013). A Survey on Recommendations in Location-based Social Networks. **Journal of ACM Transactions on Intelligent Systems and Technology** 5 (1).

Bawden, D. and Robinson, L. (2009). The dark side of information: overload, anxiety and other paradoxes and pathologies. **Journal of Information Science** 35 (2): 180-191.

Beer, D. (2008). Social network(ing) sites...revisiting the story so far: A response to danah boyd & Nicole Ellison. **Journal of Computer-Mediated Communication** 13 (2): 516-529.

Berjani, B., and Strufe, T. (2011). A recommendation system for spots in location-based online social networks. In **Proceedings of the 4th Workshop on Social Network Systems 2011** Article No. 4 (pp 1-6). NY, USA: ACM New York.

Biederman, P. S., Lai, J., Laitamaki, J. M., Messerli, H. R., Nyheim, P. D., and Plog, S. C. (2007). **Travel and Tourism: An Industry Primer.** Pearson Custom Library: Hospitality and Culinary Arts Series. Jersey, USA: Prentice Hall.

Boyd, D. M., and Ellison, N. B. (2007). Social Network Sites: Definition, History, and Scholarship. **Journal of Computer-Mediated Communication** 13 (1): 210-230.

Brücher, H., Knolmayer, G., and Mittermayer, M-A. (2002). Document Classification Methods for Organizing Explicit Knowledge. In **Proceedings of the 3rd European Conference on Organizational Knowledge, Learning, and Capabilities** 2002 (pp 1-26). Athens, Greece: University of Bern, Institute of Information Systems.

Burk, R. (2002). Hybrid Recommender Systems: Survey and Experiments. **Journal of User Modeling and User-Adapted Interaction** 12 (4): 331-370.

Cai, D. M., Gokhale, M., and Theiler, J. (2007). Comparison of feature selection and classification algorithms in identifying malicious executables. **Journal Computational Statistics & Data Analysis** 51(6): 3156-3172.

Cambridge Dictionaries Online. (2012). **social networking site** [On-line]. Available: http://dictionary.cambridge.org/dictionary/business-english/social-networking-site?q=Social+networking+site

Charland, A., and LeRoux, B. (2011). Mobile Application Development: Web vs. Native. **Communications of the ACM** 54 (5): 49-53.

Chatcharaporn, K., Angskun, J., and Angskun, T. (2011). Mobile Augmented Reality Based on Social Network. In **Proceedings of the 7th International Conference on Computing and Information Technology 2011** (pp 133-138). Bangkok, Thailand. Bangkok: KMUTNB.

Chatcharaporn, K., Angskun, J., and Angskun, T. (2012). SNSCombiner: A Module for Combining Social Networking Services to Identify User's Interests in Tourism Domain. In **Proceedings of Burapha University International Conference 2012** (pp 243-252). Pattaya, Thailand: Burapha University.

Chatcharaporn, K., Angskun, J., and Angskun, T. (2013). Improving Performance of a Mobile Personalized Recommendation Engine using Multithreading. In **Proceedings of the 10th International Joint Conference on Computer Science and Software Engineering 2013** (pp 172-178). Mahasarakham, Thailand: Mahasarakham University.

Chatcharaporn, K., Angskun, J., and Angskun, T. (2014). Tourist Attraction Categorization using a Latent Semantic Analysis and Machine Learning Techniques. **INFORMATION: An International Interdisciplinary Journal** 2014 (17).

Cheng, C., Yang, H., King, I., and Lyu M. R. (2012). Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. In **Proceedings of the 26th AAAI Conference on Artificial Intelligence 2012** (pp 17-23). Toronto, Ontario, Canada: AAAI Press.

Cheng, C., Yang, H., King, I., and Lyu M. R. (2012a). Where You Like to Go Next: Successive Point-of-Interest Recommendation. In **Proceedings of the 23rd international joint conference on Artificial Intelligence 2012** (pp 2605-2611). Palo Alto, California, USA: AAAI Press.

Cheng, R., and Vassileva, J. (2006). Design and evaluation of an adaptive incentive mechanism for sustained educational online communities. **User Modeling and User-Adapted Interaction** 16 (3-4): 321–348.

Chin, D. N. (2001). Empirical evaluation of user models and user-adapted systems. **User Modeling and User Adapted Interaction** 11 (1/2): 181–194.

Choi, C., Cho, M., Choi, J., Hwang, M., Park, J., and Kim, P. (2009). Travel Ontology for Intelligent Recommendation System. In **Proceedings of the 3rd Asia International Conference on Modelling & Simulation 2009** (pp 637-642). Washington DC, USA: IEEE Computer Society.

Chuang, C. Y., Lin, Y. B., Ren, Z. J., and Yeh, Y. T. (2011). User-Generated E-Book from Facebook Contents. In **Proceedings of the 7th International Wireless Communications and Mobile Computing Conference 2011** (pp 1918-1922). New York, USA: IEEE.

Coyle, L., and Cunningham, P. (2003). Exploiting re-ranking information in a case-based personal travel assistant. In **Proceedings of the 5th international conference on Case-based reasoning 2003** (pp 11-20). Berlin, Heidelberg, Germany: Springer.

Deerwester, S., Dumais, S., Furnas, G.W., Landauer, T.K., and Harshman, R. (1990). Indexing by Latent Semantic Analysis. **Journal of the Society for Information Science** 1990 (41): 391-407.

Dictionary.com. (2012). **Social Networking Site** [On-line]. Available: http://dictionary.reference.com/browse/Social+Networking+Site?s=t

Drachsler, H., Hummel, H., and Koper, R. (2007). Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. In **Proceedings of the 1st Workshop on Social Information Retrieval for Technology-Enhanced Learning & Exchange 2007** (pp 18-26). Crete, Greece: CEUR.

Dumis, S. T., Fumas, G. W., Landauer, T. K., Deerwester, S., and Harshman, R. (1988). Using Latent Semantic Analysis to Improve Access to Textual Information. In **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems 1988** (pp 217-285). NY, USA: ACM New York.

Egger, R., and Buhalis, D. (2008). **eTourism Case Studies Management and Marketing Issues.** Oxford, UK: Butterworth-Heinemann.

Ertöz, L., Steinbach, M., and Kumar, V. (2002). A New Shared Nearest Neighbor Clustering Algorithm and its Applications. In **Proceedings of the Workshop on Clustering High Dimensional Data and its Applications at 2nd SIAM International Conference on Data Mining 2002** (pp 1-4). Arlington, VA, USA: SIAM.

Esslimani, I., Brun, A., and Boyer, A. (2009). From Social Networks to Behavioral Networks in Recommender Systems. In **Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining 2009** (pp 143-148). Washington, DC, USA: IEEE Computer Society.

Fijałkowski, D., and Zatoka R. (2011). An architecture of a Web recommender system using social network user profiles for e-commerce. In **Proceedings of the Federated Conference on Computer Science and Information Systems 2011** (pp 287-290). Szczecin, Poland: IEEE Computer Society Press.

Fink, J., and Kobsa, A. (2002). User Modeling for Personalized City Tours. **Artificial Intelligence Review** 18 (1): 33-74.

García-Crespo, A., Chamizo, J., Rivera I., Mencke M., Colomo-Palacios, R., and Gómez-Berbís, J. M. (2009). SPETA: Social pervasive e-Tourism advisor. **Journal of Telematics and Informatics** 26(3): 306–315.

Gavalas, D., and Kenteris, M. (2011). A web-based pervasive recommendation system for mobile tourist guides. **Personal and Ubiquitous Computing** 15 (7): 759-770.

Graham, W. (2008). **Facebook API Developers Guide (Firstpress)**. Apress.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The WEKA Data Mining Software: An Update. **SIGKDD Explorations** 11 (1): 10-18.

Hall, M. A., and Smith, L. A. (1998). Practical Feature Subset Selection for Machine Learning. In **Proceedings of the 21st Australian Computer Science Conference 1998** (pp 181-191). Berlin: Springer.

He, J., and Chu, W. W. (2010). A Social Network-Based Recommender System (SNRS). **Data Mining for Social Network Data** 2010 (12): 47-74.

Hillard, D. (1996). Topic Classification for Conversational Speech using Support Vector Machines and Latent Semantic Analysis. **Working Paper**.

Hristova, N., O'Hare, G. M. P., and Lowen, T. (2003). Agent-based ubiquitous systems: 9 lessons Learnt. In **Workshop on System Support for Ubiquitous Computing 5th International Conference on Ubiquitous Computing.** NY, USA: ACM New York.

Hsu, F-M., Lin, Y-T., and Ho, T-K. (2012). Design and implementation of an intelligent recommendation system for tourist attractions: The integration of EBM model, Bayesian network and Google Maps. **Journal of Expert Systems with Applications** 39 (3): 3257-3264.

Huang, Y. (2011). A Latent Semantic Analysis-Based Approach to Geographic Feature Categorization from Text. In **Proceedings of the 5th IEEE International Conference on Semantic Computing 2011** (pp 87-94). Palo Alto, CA, USA: IEEE.

Huang, Y. and Bian, L. (2009). A Bayesian network and analytic hierarchy process based personalized recommendations for tourist attractions over the Internet. **Journal of Expert Systems with Applications** 36 (1): 933-943.

Inrak, P., and Sinthupinyo, S. (2010). Applying Latent Semantic Analysis to Classify Emotions in Thai Text. In **Proceedings of the 2nd International Conference on Computer Engineering and Technology 2010** (pp V6-450-V6-454). Chengdu, China: IEEE.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. **New Phytologist** 11 (2):37-50.

Joachims, T. (1998). Text Categorization With Support Vector Machines: Learning with Many Relevant Features. In **Proceedings of the 10th European Conference on Machine Learning 1998** (pp 173-142). London, UK: Springer-Verlag.

Kabassi, K. (2010). Personalizing recommendations for tourists. **Telematics and Informatics** 27(1): 51-66.

Kamar, A. (2003). Mobile Tourist Guide (m-ToGuide). **Deliverable 1.4, Project Final Report. IST-2001-36004.**

Kanellopoulos, D. N. (2008). An ontology-based system for intelligent matching of travellers' needs for Group Package Tours. **International Journal of Digital Culture and Electronic Tourism.** 1 (1): 76-99.

Khan, A., Baharudin, B., Lee, L. H., and Khan, K. (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. **Journal of Advances in Information Technology** 1(1): 4-20.

Kim, K-J., and Ahn, H. (2012). Hybrid Recommender Systems using Social Network Analysis. **World Academy of Science, Engineering and Technology** 2012 (64): 879-882.

Kim, W. (2002). Personalization: Definition, status, and challenges ahead. **Journal of Object Technology** 1 (1): 29-40.

Ko, M. N., Cheek, G. P., and Shehab, M. (2010). Social-Networks Connect Services. **Journal of Computer** 43 (8): 37-43.

Koychev, I. (2000). Gradual Forgetting for Adaptation to Concept Drift. In **Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning 2000** (pp 101-106). Berlin, Germany.

Kurashima, T., Iwata, T., Irie, G., and Fujimura, K. (2010). Travel route recommendation using geotags in photo sharing sites. In **Proceedings of the 19th ACM international conference on Information and knowledge management 2010** (pp 579-588). NY, USA: ACM New York.

Landauer, T. K., Foltz, P. W., and Laham, D. (1998). Introduction to Latent Semantic Analysis. **Discourse Processes**. 25 (2-3): 259-284.

Lee, C-S., Chang, Y-C., and Wang M-H. (2009). Ontological recommendation multi-agent for Tainan City travel. **Expert Systems with Applications: An International Journal.** 44 (1): 6740-6753.

Lewis, D. D. (1998). Naive (Bayes) at Forty: The Independence Assumption in Information Retrieval. In **Proceedings of the 10th European Conference on Machine Learning 1998** (pp 4-15). London, UK: Springer-Verlag.

Lian, D., and Xie, X. (2011). Collaborative activity recognition via check-in history. In **Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks 2011** (pp 45-48). NY, USA: ACM New York.

Loni, B., Khoshnevis, S. H., and Wiggers, P. (2011). Latent Semantic Analysis for Question Classification with Neural Networks. In **Proceedings of IEEE workshop on Automatic Speech Recognition and Understanding 2011** (pp 437-442). Waikoloa, HI, USA: IEEE.

Lv, L., and Lui, Y-S. (2005). Research of English Text Classification Methods based on Semantic Meaning. In **Proceedings of ITI 3rd International Conference on Information and Communications Technology: Enabling Technologies for the New Knowledge Society 2005** (pp 689-700). Cairo, Egypt: IEEE.

Ma, H., Zhou, D., Liu, D., Lyu, M. R., and King, I. (2011). Recommender systems with social regularization. In **Proceedings of the 4th ACM international conference on Web search and data mining 2011** (pp 287-296). New York, USA: ACM.

McCallum, A., and Nigam, K. (1998). A comparison of event models for Naive Bayes text classification. In **AAAI-98 Workshop on Learning for Text Categorization 1998** (pp 41-48). Dortmund, Germany: AAAI Press.

Melville, P., and Sindhwani, V. (2010). Recommender Systems. **Encyclopedia of Machine Learning 2010** (pp 829-838). Claude Sammut and Geoffrey Webb (Eds), Springer.

Meteren, R. V., and Someren, M. V. (2000). Using Content-Based Filtering for Recommendation. In **Proceedings of the MLnetECML2000 Workshop Machine Learning in the New Information Age** 2000 (pp 47-56). London, UK: Springer-Verlag.

Miao, D., Duan, Q., Zhang, H., and Jiao, N. (2009). Rough set based hybrid algorithm for text classification. **Expert Systems with Applications: An International Journal archive** 36 (5): 9168-9174.

Micarelli, A., and Sciarrone, F. (2004). Anatomy and Empirical Evaluation of an Adaptive Web-Based Information Filtering System. **User Modeling and User-Adapted Interaction** 14 (2-3): 159-200.

Minas, G., and Soldo, F. (2008). Exploring collaborative filters: Neighborhood-based approach. **Working Paper** 2008 (pp 1-7). Austin, Texas, USA: Department of MSIS, University of Texas.

Mobasher, B. (2007). Data Mining for Web Personalization. **The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science** 4321, 90-135.

Montaner, M., López, B., and de la Rosa, J. L. (2003). A taxonomy of recommender agents on the internet. **Artificial Intelligence Review.** 19 (4): 285-330.

Myllymaki, P., and Tirri, H. (1993). Bayesian Case-Based Reasoning with Neural Network. In **Proceedings of the IEEE International Conference on Neural Network 1993**(1) (pp 422-427). San Francisco, CA, USA: IEEE.

Ng, H. T., Goh, W. B., and Low, K. L. (1997). Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization. In **Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval 1997** (pp 67-73). NY, USA: ACM New York.

Nielsen, J. (2000). **Why You Only Need to Test with 5 Users** [On-line]. Available: http://www.useit.com/alertbox/20000319.html

O'Grady, M. J., and O'Hare, G. M. P. (2004). Gulliver's Genie: Agency, Mobility, Adaptivity. **Computers & Graphics** 28(5): 677-689.

Ou, S., Pekar, V., Orasan, C., Spurk, C., and Negri, M. (2008). Development and alignment of a domain-specific ontology for question answering. In **Proceedings of the 6th Edition of the Language Resources and Evaluation Conference 2008** (pp 2221-2228). La Valletta, Malte: European Language Resources Association (ELRA).

Oxford Dictionaries. (2012). **social networking** [On-line]. Available: http://oxforddictionaries.com/definition/english/social%2Bnetworking?q=social+networking

Pazzani, M. (1999). A framework for collaborative, content-based and demographic filtering. **Artificial Intelligence Review** 13 (5-6): 393-408.

Picot-Clémente, R. and Bothorel, C. (2013). Recommendation of shopping places based on social and geographical influences. In **Proceedings of the 5th ACM RecSys Workshop on Recommender Systems and the Social Web 2013**. Hong Kong, China: CEUR-WS.org.

Porter, M. F. (1980). An algorithm for suffix stripping. **Program: electronic library and information systems** 14 (3): 130 - 137. Bradford, West Yorkshire, UK: MCB UP Ltd.

Quinlan, J. R. (1986). Induction of Decision Trees. **Journal of Machine Learning**. 1 (1): 81-106.

Quinlan, J.R. (1993). **C4.5: Programs for Machine Learning**. San Francisco, USA: Morgan Kaufmann Publishers.

Rahimi S. M., and Wang, X. (2011). Location Recommendation Based on Periodicity of Human Activities and Location Categories. **Advances in Knowledge Discovery and Data Mining** 2013 (2): 377-389. Berlin, Heidelberg, Germany: Springer.

Rao, K. N., and Talwar, V. G. (2008). Application Domain and Functional Classification of Recommender Systems—A Survey. **DESIDOC Journal of Library & Information Technology** 28 (3): 17-35.

Resnick, P., and Varian, H. (1997). Recommender Systems. **Communications of the ACM** 40(3): 56-58.

Rich, E. (1983). Users are individuals: individualizing user models. **International Journal of Man-Machine Studies** 18 (3): 199-214.

Rich, E. (1989). Stereotypes and user modeling. **User Models in Dialog Systems**. Berlin, Heidelberg, Germany: Springer.

Rich, E. (1999). Users are individuals: individualizing user models. **International Journal of Human–Computer Studies** 51 (2): 323-338.

Richmond, R. (2010). **Three Best Ways to Use Location-Based Social Media** [On-line]. Available: http://online.wsj.com/article/SB10001424052748703597204575483832278936028.html

Ruiz, M. E., and Srinivasan, P. (1998). Automatic Text Categorization Using Neural Network. In **Proceedings of the 8th ASIS SIG/CR Workshop on Classification Research 1998** (pp 59-72). Silver Spring, MD: American Society for Information Science.

Salton, G., and McGill M. J. (1987). **Introduction to Modern Information Retrieval**. NY, USA: McGrawHill.

Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based Collaborative Filtering Recommendation Algorithms. In **Proceedings of the 10th international conference on World Wide Web 2001** (pp 285-295). NY, USA: ACM New York.

Schafer, J. B., Konstan, J. A., and Riedl, J. (1999). Recommender systems in e-commerce. In **Proceedings of the 1st ACM Conference on Electronic Commerce 1999** (pp 158-166). NY, USA: ACM New York.

Schafer, J. B., Konstan, J. A., and Riedl, J. (2000). Electronic commerce recommender applications. **Journal of Data Mining and Knowledge Discovery** 5 (1-2): 115-152.

Schafer, J. B., Frankowski, D., Herlocker, J., and Sen S. (2007). Collaborative filtering recommender systems. In P. Brusilovsky, A. Kobsa and W. Nejdl (eds.). **The Adaptive Web: Methods and Strategies of Web Personalization** (pp 291-324). Berlin, Heidelberg, Germany: Springer.

Schiaffino, S., and Amandi, A. (2009). Building an expert travel agent as a software agent. **Journal of Expert Systems with Applications** 36 (2): 1291-1299.

Shih, D-H., Yen, D. C., Lin, H-C., and Shih M-H. (2011). An implementation and evaluation of recommender systems for traveling abroad. **Journal of Expert Systems with Applications** 38 (12): 15344-15355.

Sokolava, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. **Information Processing and Management** 45 (4): 427-437.

Srivihok, A., and Sukonmanee, P. (2005). Intelligent Agent for e-Tourism: Personalization Travel Support Agent using Reinforcement Learning. In **Proceedings of the Fourteenth International World Wide Web Conference 2005**. NY, USA: ACM New York.

Sunden, J. (2003). **Material Virtualities: Approaching Online Textual Embodiment**. New York: Peter Lang.

Vapnik, V. N. (1995). **The Nature of Statistical Learning Theory**. NY, USA: Springer.

Vasconcelos, M. A., Ricci, S., Almeida, J., Benevenuto, F., and Almeida, V. (2012). Tips, dones and todos: uncovering user profiles in foursquare. In **Proceedings of the 5th ACM international conference on Web search and data mining 2012** (pp 653-662). NY, USA: ACM New York.

Vozalis, M., and Margaritis, K. G. (2004) Collaborative Filtering Enhanced by Demographic Correlation. **In Proceedings of the AIAI Symposium on Professional Practice in AI, part of the 18th World Computer Congress 2004** (pp 393-402). Toulouse, France.

Wan, Y., and Tong, H. (2008). Categorization and Monitoring of Internet Public Opinion Based on Latent Semantic Analysis. In **Business and Information Management, 2008. ISBIM '08. International Seminar on 2008** (2): 121-124. Wuhan, China: IEEE.

Wang, H., Terrovitis, M., and Mamoulis, N. (2013). Location recommendation in location-based social networks using user check-in data. In **Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems 2013** (pp 374-383). NY, USA: ACM New York.

Wang, J, De Vries, A.P., and Reinders, M. J. T. (2008). Unified relevance models for rating prediction in collaborative filtering. **ACM Transactions on Information Systems** 26 (3): 1-42.

Wang, W., Zeng, G., Zhang, D., Huang, Y., Qiu, Y., and Wang, W. (2008). An Intelligent Ontology and Bayesian Network based Semantic Mashup for Tourism. In **Proceedings of International Conference on Services Computing 2008** (pp 198-201). Washington, DC, USA: IEEE Computer Society.

Wang, X., and Zheng, Q. (2013). Text Emotion Classification Research Based on Improved Latent Semantic Analysis Algorithm. In **Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering 2013**. Paris, France: Atlantis Press.

Wang, Z-Q., Sun, X., Zhang, D-X., and Li, X. (2006). An Optimal SVM-Based Text Classification Algorithm. In **Proceedings of the 5th International Conference on Machine Learning and Cybernetics** 2006 (pp 13-16). August, Dalian, China: IEEE Press.

Weiss, S. M., and Indurkhya, N. (2001). Lightweight Collaborative Filtering Method for Binary Encoded Data. In **Proceedings of the 5th European Conference on Principles and Practice of Knowledge Discovery in Databases 2001** (pp 484-49). Berlin, Heidelberg, Germany: Springer.

Williamson, S., and Ghahramani, Z. (2008). Probabilistic models for data combination in recommender systems. In **Proceedings of the NIPS Workshop on Learning from Multiple Sources 2008** (pp 1-15). Whistler, Canada: MIT Press.

Xiao, X., Zheng, Y., Luo, Q., and Xie, X. (2010). Finding similar users using category-based location history. In **Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems 2010** (pp 442-445). NY, USA: ACM New York.

Xu, G., Zhang, Y. and Zhou, X. (2005). Towards User Profiling for Web Recommendation. In **Proceedings of the 18th Australian Joint Conference on Artificial Intelligence 2005** (pp 415-424). Berlin, Heidelberg, Germany: Springer.

Yang, Y., and Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. In **Proceedings of the 14th International Conference on Machine Learning 1997** (pp 412-420). San Francisco, CA, USA: Morgan Kaufmann Publishers.

Yap, G-E., Tan, A-H., and Pang, H-H. (2005). Dynamically-optimized context in recommender systems. In **Proceedings of the 6th International Conference on Mobile Data Management 2005** (pp 265-272). NY, USA: ACM New York.

Ye, M., Yin, P., and Lee, W-C. (2010). Location recommendation for location-based social networks. In **Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems 2010** (pp 458-461). NY, USA: ACM New York.

Ye, M., Yin, P., Lee, W-C., and Lee, D-L. (2011). Exploiting geographical influence for collaborative point-of-interest recommendation. In **Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval 2011** (pp 325-334). NY, USA: ACM New York.

Ying, J. J-C., Lu, E. H-C., Kuo, W-N., and Tseng, V. S. (2012). Urban point-of-interest recommendation by mining user check-in behaviors. In **Proceedings of the ACM SIGKDD International Workshop on Urban Computing 2012** (pp 63-70). NY, USA: ACM New York.

Yu, B., Xu, Z-B., and Li, C-H. (2008). Latent Semantic Analysis for Text Categorization using Neural Network. **Knowledge-Based Systems**, 21(8): 900–904.

Yuan, Q., Cong, G., Ma, Z., Sun, A., and Magnenat-Thalmann, N. (2013). Time-aware point-of-interest recommendation. **In Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval 2013** (pp 363-372). NY, USA: ACM New York.

Zhang, J-D., and Chow, C-Y. (2013). iGSLR: personalized geo-social location recommendation: a kernel density estimation approach. In **Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems 2013** (pp 334-343). NY, USA: ACM New York.

Zheng, N., Jin, X., and Li, L. (2013). Cross-region collaborative filtering for new point-of-interest recommendation. In **Proceedings of the 22nd international conference on World Wide Web companion 2013** (pp 45-46). Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Zheng, Y., Zhang, L., Xie, X., and Ma, W-Y. (2009). Mining interesting locations and travel sequences from GPS trajectories. In **Proceedings of the 18th international conference on World Wide Web 2009** (pp 791-800). NY, USA: ACM New York.

Zheng, Y., Zhang, L., Ma, Z., Xie, X., and Ma, W-Y. (2011). Recommending friends and locations based on individual location history. **Journal of ACM Transactions on the Web** 5 (1): 1-44.

Zheng, Y., and Zhou, X. (2011). Location-Based Social Networks: Users. **Computing with Spatial Trajectories** (243-276)**.** NY, USA: Springer.

CURRICULUM VITAE

Mr. Komkid Chatcharaporn was born on March 29, 1986 in Surin Province, Thailand. He received Bachelor of Information Science from Suranaree University of Technology, Thailand in 2007. In 2010, he has got a scholarship from Institute of Social Technology, School of Information Technology, Suranaree University of Technology to pursue his doctoral degree in Information Technology Program at Suranaree University of Technology. His major research interests are social networking services, text categorization and recommendation system.