國立臺灣大學電機資訊學院資訊工程學研究所

碩士論文

Department of Computer Science and Information Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

以本體論爲基礎之
情境感知校園景點推薦系統

An Ontology-based
Context-aware Recommender System
for Campus Scenic Spots

陳筱薇
Hsiao-Wei Chen

指導教授：許永眞 博士
Advisor: Jane Yung-jen Hsu, Ph.D.

中華民國九十八年六月
June, 2009

Acknowledgments

感謝所有陪伴我成長的人，你們的鼓勵讓這篇論文順利完成。首先感謝我的指導教授，許永真老師，謝謝老師讓我學到做研究的方法與態度，並適時的給予我指導，這一段研究的生活很充實，收穫很多。感謝我的口試委員：朱浩華教授、王傑智教授，給予我寶貴的建議，讓我的論文更完整。

研究過程中有不少的困難與壓力，感謝Context Aware 組的伙伴們對我的包容與鼓勵。謝謝翰文認眞的陪我討論、解決問題;謝謝栗子不厭其煩的聽我訴苦;謝謝啓嘉一言道出我的盲點;謝謝元翔告訴我要有信心。謝謝婉容學姐以及理律學長將研究做的很完整，讓我接手時容易許多。在研究上受到你們很多的幫助，眞得謝謝你們了！

感謝實驗室的大家：婉容學姐、小嫺、小朋友、摸貓、David、好圓、忠毅。有了你們336多了許多歡笑，就像一個大家庭，很溫暖。感謝每一位幫助我做實驗的學弟妹：谷原、世強、彥伶、守壹、leeaa、俊豪、居正。特別感謝台大訪客中心的松翰學長，因爲你們的熱心協助，杜鵑花節的活動才能辦的有聲有色。

最後謝謝義翔一直陪伴著我克服種種的困難，給我前進的力量;謝謝母親對我長久的呵護，您是我最重要的存在。

Abstract

In this paper, we propose a context-aware campus spots recommender system which detects the changes in the environment, and provides a list of spots to user whose preference cannot be retrieved at the trip beginning, on the basis of user's responses of what had visited in the trip, what time it is, whether it rains, who company with the visitor, and attractions information. The traditional recommender systems overlooked that a decison making differs in different context (location, time, or weather). In order to get high quality recommendations, in the general recommender system user has to rate a sufficient number of items; However, this system learns user's preferences during the trip and recommends spots that user are interested in, even if this is his first time contact with the system.

The system uses three main technologies: knowledge conceptualization, context-awareness ability, dynamic recommneder algorithm. Ontology is used to represent a set of concepts within the tourist domain. A spatial ontology organize spots information, and conceptualize geographic knowledge of National Taiwan University (NTU) campus, such as Fu Bell is a spot, Royal Palm Blvd. is a road, and Fu Bell is on Royal Palm Blvd. is a geographic knowledge. The temporal concepts are built in ontology which could infer high-level information with the raw data. For example, 9:00 AM is a time stamp, by inference the system obtain that 9:00 AM is in the Morning and it

belongs to eating time. Context-awareness ability copes with the changes in the enviroment. In short, the visitors could experience recommendations depending on their personal data and the environment conditions. With up-to-date user's responses in the trip, the system dynamicly provides recommendations which vary with different time and weather condtion.

摘要

情境感知校園景點推薦系統根據參觀者的基本資料、參觀的行爲、環境中的資訊(時間、天氣)以及景點間的相互關係，動態地推薦景點。傳統景點推薦系統只考慮使用者偏好以及景點資訊，並沒有將情境的變化納入考量。此外，本系統針對新的使用者(new user)做推薦，在行程開始之前，系統並不假設擁有使用者過去的旅遊偏好資訊。在使用者遊玩的過程中，系統觀察使用者與系統的互動，參考使用者對拜訪過的景點的喜好度，同時察覺作用中環境的變化，適時產生景點推薦。換句話說，本論文所提出的情境感知景點推薦系統，其需求和推薦系統不同，傳統產品推薦系統，會有一段冷開機(cold-start)的時間，這類推薦系統的品質要隨著對使用者的了解而增加，也就是說，它們必須累積了大量使用者的歷史交易紀錄後，才能得到有品質保證的推薦。但是本景點推薦系統，力求能在袓時間內察覺使用者偏好，推薦環境中使用者有興趣的景點，即便這個使用者是第一次使用本系統。

本系統包含三個主要技術：知識的具體化、情境感知、動態推薦。知識的具體化使用本體論(Ontology)在特定應用領域表達一些概念的集合本論文分別建立了時間和空間的知識庫。空間本體論用來架構景點在地理空間上的概念，例如：傅鐘是一個景點，椰林大道是一條路，傅鐘在椰林大道上是一個知識。

在時間本體論上，根據本體論以節點爲主的建構概念，使具體化的功能得以從初步資料推論高階的資訊。例如：九點是一個時間標記(time stamp)，透過推論讓電腦知道，九點在早上而且屬於吃飯的時間。情境感知(context-aware)處理環境因子的變化，例如：使用者所在地、時間、天氣等資訊。本推薦系統根據使用者的基本資料、同伴、因應現在的時間、天氣的變化，動態調整推薦的內容。

本論文實作了二個階段的實驗：第一階段在台大杜鵑花節我們邀請民眾考慮時間、天氣、以及與誰一同來玩等因素，爲校園景點打分數。透過此活動收集到93筆移動軌跡，949筆景點喜好紀錄，以此建立使用者導向地理模型(user-based location model)，強化原有的景點模型做爲推薦的基礎知識。在此地理模型中除了紀錄景點基本資料還包括從分析民眾在台大遊玩的移動軌跡，發現的景點之間的關係(例如：我們觀察到很多人看完傅鐘接下來就會去總圖，因此這二個景點之間產生”下一站”的關係，即傅鐘的下一站是總圖)。第二階段讓使用者帶著本推薦系統實際走訪校園，評估本系統的接受度、實用性。實驗進行方式，系統同時交錯列出二個推薦的景點列表，其中一個是有將情境的變化納入考慮，另一個則無，而使用者並不知道它們的差別。我們紀錄十位使用者對推薦結果的反應(接受、拒絕、目前沒意見)，以此評估情境因子對景點推薦的助益。

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we describe the motivation of this ontology-based context-aware campus spots recommender system. Furthermore, we menstion the limitations of common context aware applications and recommendation algorithm to point out the purpose of this work.

1.1 Motivation

People travel for different reasons including for business, for education, for leisure, and for attending conference. It is useful if there exist a recommender system which provides a flexiable and personalized recommendation to anyone who travel but may not be able to plan the tour in advance. In this paper, we propose a campus scenic spots recommender system which can provide the novel user with a list of spots which are likely of interest to this new user on the basis of the user's preference, what time it is, whether it rains, and who company with the visitor.

1.2 General Background Information

In this work, we uses three main technologies: knowledge conceptualization, context-awareness ability and dynamic recommneder algorithm. Ontology is used to represent a set of concepts within the tourist domain, context-awareness ability copes with the changes in the enviroment, and dynamic recommnedations vary with current time and weather conditon.

1.2.1 Recommender System

The *recommender systems* [1] was defineed as a specific type of Information Filtering (IF) technique that attempts to present items (movies, music, books, news, images, web pages, *etc.*) that are likely of interest to the user. Recently, due to the information overload recommender systems are getting attentions not only from academic research but also from e-commerce sites including Amazon.com [2], moviepilot.de [3], and musicovery [4]. Two basic approaches have emerged for making recommendations: content-based approach and collaborative filtering approach. The Content-based filtering analyzes the content of items (e.g., web pages) that have been rated to create a user's interest profile in terms of regularities in the content of information that was rated highly. This

[1]source: `http://en.wikipedia.org/wiki/Recommender_system`
[2]source: `http://www.amazon.com/`
[3]source: `http://www.moviepilot.de/`
[4]source: `http://www.musicovery.com/`

profile may be used to rate other unseen items or to construct a query of a search engine. On the contrary, the collaborative approaches find information sources for an individual that have been rated highly by other users whose ratings pattern is similar to that of the user.

1.2.2 Context-Aware Computing

The free on-line Dictionary of Computing defines *Context* is that which surrounds, and gives meaning to something else. Context-Aware applications are aware of the context in which they are run, such context-aware software adapts according to the location of use, the collection of nearby people, and accessible devices. A system with these capabilities can examine the computing environment and react to changes to the environment with no user typing into a computer. In other words, when computers can sense the physical world, people can dispense with much of what is done with keyboards and mice. In this paper, some important aspects of context are: where the visitor is, what time it is, whether it rains, how visitor feel about current trip, and who the visitors is with.

1.3 Context-Aware Spot Recommender System

A fairly large body of literature exists on recommendation algorithm and context aware application ; however, both of them have various limitations. Context-aware applications are unable to predict a user's preference in an unseen situation because the rules are static. The existing recommendation algorithms could not model complex situation of which recommender space exceeds two dimensions. The traditional recommender systems overlooked that a decison making differs in different context. For example, when it rains, most people do not engage in outdoor activities and when it approach meal time people will find place support food. In this work, context-aware technology detects changes in the environment to reach these goals.

1.3.1 Purpose

In light of above concerns, we present a ontology-based context-aware campus spot recommender system which give recommendations to novel users whose preference cannot be retrieved at the trip beginning based on user's responses of what had visited in this trip, contextual information and attractions profile. Additionally, from the view of visitor, this context-aware recomnneder system has three benefits:

1. User does not need to plan trip in advance.
2. The system actively suggests spots to user based on different context.
3. The recommendation satisfies the specific interests of individuals, rather than the majority.

1.3.2 System Archetecture

Here, a trip is divided into three period: Pre-trip, In-trip ,and Post-trip. The system works at the In-trip period (see figure 1.1). At a trip begging, we have no historical tour records about the new user. After the user visiting and rating for some landmarks, the

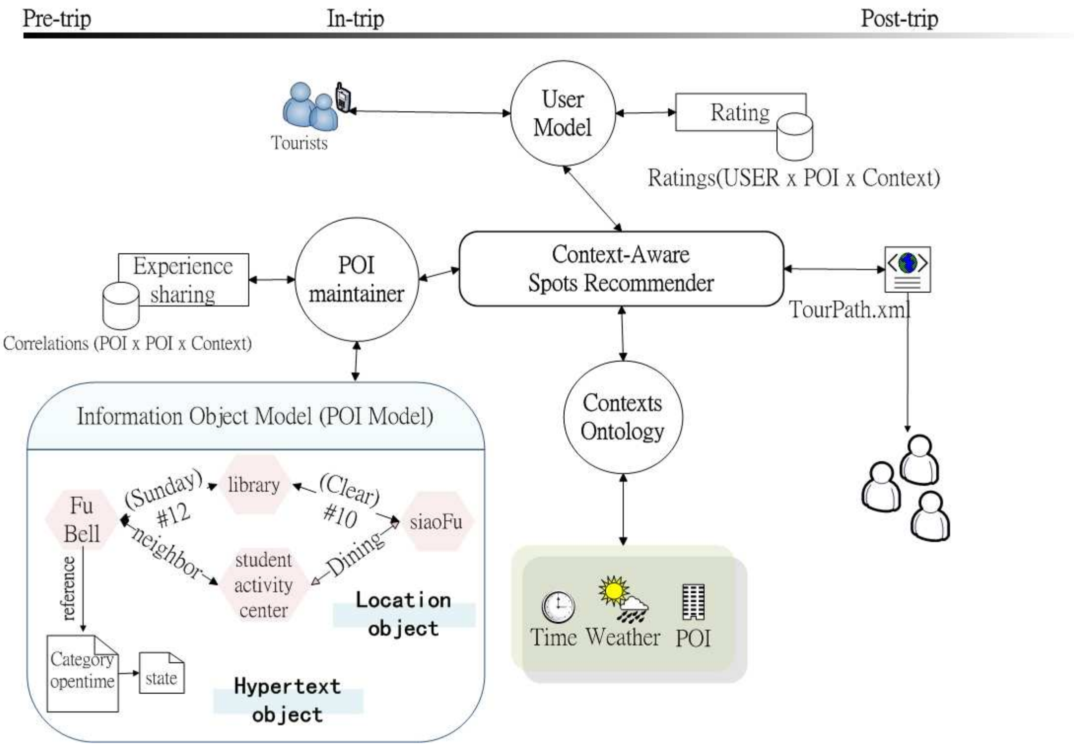

Figure 1.1: System Architecture

predictor makes recommendations for unknown spots on the basis of demographics, visiting habit of user, current context and location attributions. In brief, the input of the system includes:

- **Location model** contains properties of spots such as geographical position, category, services, *etc.*, which were predefined in spatial ontology and latent relationships among POIs which learned by analyzing moving patterns of visitors over NTU campus.

- **User model** considers the demographics and visiting behavior of user in the trip. The demographic information (e.g., age, gender, and companies) are obtained from the welcome screen. Observing users interact with the system, we design an implicit rating mechanism.

- **Context model** descries the temporal conception and weather conditions(e.g., clear, cloudy, and rainy) .

The output is a context-aware personalized list of scenic spots.

1.4 Organization

The remainder of this paper is organized as follows: Chapter 2 describes the prior work. The problem formulation and details approach are presented in Chapter 3. Following the definition, Chapter 4 briefly describes the implementation and experiment results. Finally, concluding remarks are stated in Chapter 5.

Chapter 2

Background and Related Work

In this chapter, we review some prior work on recommneder system, context-aware application, user modeling, and ontology. Firstly, we introduce the state of the art recommender algorithms, in order to improve recommendation capabilities and make recommender systems applicable to broader range of applications, more work focus on the possible extensions to the current recommendation methods. Next, we investigate how to identify the user's interests unobtrusively by an improvement of understanding of users [1][26]. Then, we shift attention to context-awareness technology, and then incorporate contextual information into the recommendation process [40][9][3]. Finally, ontology is the key to make the context-awareness come true. we talks about some famous projects have been designed ontology used for knowledge sharing. And in this work, we used ontology to describe the geospatial and temporal knowledge about landmarks on NTU campus.

2.1 Recommender System

Recommender systems is a kind of learning systems that predict the utility towards new items regarding a particular user based on user's explicit or implicit ratings, others' opinions, and the attributes of user and items. Recommender systems emerged as an independent research area in the mid-1990s when researchers started focusing on recommendation problems that explicitly rely on the ratings structure. Typically we do not have a complete set of votes across all items, regardless of the type of vote data available, recommendation algorithms must address the issue of missing data. The users' preferences are learning target function, we have reasons to believe that there exists some other target functions in the dataset that consistently behaves similar, neutral or opposite to the target function for the particular user. Rating-based recommendation algorithms rely on an interpretation that any vote indicates a positive preference [38]. In a case of a spot recommender system, Alice assigns a rating of 4 (out of 5) for FullBell, and also votes 2 for Library, then we can say that Alice probably like fullBell more than Library.

The recommendation methods are usually classified into three main categories based on how recommendations are made [1]: *content-based*, *collaborative filtering*, and *hybrid*.We introduce them and describe which and why we used in this work.

- **Content-based recommendations:** The user will be recommended items which have higher degree of similarity to those he preferred in the past.

- **Collaborative recommendations:**

  - **user-based:** The user will be recommended items that people with similar tastes liked in the past.

  - **item-based[21][31][23]:** The similarities among items were computed based on users' responses not their content.

- **Hybrid approaches:** These methods combine content-based and collaborative methods.

2.1.1 Content-Based Recommendation

The content-based approach roots in information retrieval, it analyzes the description of items that have been rated by the same user and the description of items to be recommended. Many content-based approaches focus on recommending text-based items like documents, a document is represented as a set of most informative words which characterizing it extracting from its content, i.e., the words that are more associated with one class of documents than another, and use those keywords to determine the appropriateness of the document for recommendation purposes. One of the well-known measures for specifying keyword weights in Information Retrieval is the *term frequency/inverse document frequency (TF-IDF)*. In fact, the similarity among items are computed according to their content.

Content-based approachs have their own limitations. First, the content analysis capability is limited by features that are explicitly associated with items. In order to have a sufficient set of features, the content either in a form that can be parsed automatically by a computer (e.g., text).In this work, we use ontology to build the location model in which each spot not only has its basic attributes, including identification, category, position, *etc.* But each spot also has some user oriented properties like popular time to be visited which was obtained by observing users' routes in the database, not the content of spots. Second, in order to get high quality recommendations the content-based recommender system has to really understand the user's preferences; means that the user has to rate a sufficient number of items. However it is not a practical assumption here, in this work, we give recommendations to users whose preference cannot be retrieved at trip beginning, that is to say the content-based approach would be unable to get accurate recommendations to the user who has very few rating records. This is called a new user problem, and we alleviate it by assigning the new user into a high level group according his demographical features.

2.1.2 Collaborative Filtering Recommendation

Collaborative filtering recommender systems predict the rate of items to a particular user based on a collection of ratings rated by other like-minded people. Unlike content-based recommendation methods, typically, collaborative filtering does not use any information regarding the actual content of items, but rather match the preference patterns of active user to those of other users. That is why it called collaborative filtering or social recommender system.

Collaborative filtering has been shown as one successful recommender system technology in many practical applications, but here we describe some potential problems associated with correlation-based collaborative filtering models. First, the performance of recommendations made by collaborative filtering degrades with the number of customers and items, it is called large scale problems. Sarwar *et al* use association rule technique to produce top-N recommendations to address large-scale purchased and preference data [32], and they also explore a technology called Singular Value Decomposition (SVD) to reduce the dimensionality of recommender system databases to quickly produce high quality recommendations, even for very large scale problems [33]. Another problem occurs when the different product names refer to the similar objects. Correlation based recommender systems do not discovery this latent association and treat these products exactly differently. For example, considering two visitor one of them rates low-fat yougurt as ”high” and the other rates low-fat milk as ”high”. Correlation based approach would see no match between products to compute correlation and would be unable to find the latent association that both of them like low-fat dairy products. In our work, both large scale and synonymy problems may relieve, since the visitor rates for a spot represented by its unique position, and location ontology is used for modeling the latent relationships among spots. Again the number of spots in NTU campus is fixed and the number of visitors grows slowly, the large scale problem may not serious.

Usually, in any recommender application, the number of ratings already obtained is usually very small compared to the number of ratings that need to be predicted. Effective prediction of ratings from a small number of examples is important. User-based collaborative methods look for similar users’ opinions to make predictions, more precisely, the ratings pattern of individuals are used to determine similarity. Typically, user-based collaborative filtering systems find like-minded customers (also called neighborhood) by *Pearson correlation* or *cosine-based similarity* between the opinions of the users. These statistical approaches find the neightborhood of the active user, i.e., others who have similar tastes in items (rate the same items similarly), such a correlation is most meaningful when there are many objects rated in common between users. Unfortunately, we may not expect there to be a large number of ratings in this work. What's worse the nearest-neighbor algorithms rely upon exact matches, this meant that the correlation is only defined between individuals who have rated at least two items in common, whenever there are relatively few user-specified ratings, for either the active user or the neighbors, not many pairs of users have correlation at all. The situation that there is fairly little ratings data were included is called sparsity problem, and it causes collaborative methods might be expected to fail, and also the accuracy of recommendations may be poor. Many efforts have been made to overcome the sparsity problems of a purely collaborative approach: Sarwar *et al*.incorporated semi-intelligent filtering agents [34] into system. These agents evaluated and rated each product, using syntactic features, but the filterbot is only for text-based item since the rate is gotten by content analysis. Breese *et al*.assume some default value [8] as rating for the missing data. The dimensionality reduction [33] is a different approach for dealing with sparsity, through content analysis and reduce item space by using k feature to describe n items, where n is large and k less than n to increase density and thus find more ratings. Providing a dense ratings set helped alleviate coverage and improved quality, however, this kind of solution did not address the fundamental problem of poor relationships among like-minded but sparse-rating customers.

As was observed, sparsity poses a computational challenge, nearest-neighbor algorithms become harder to find neighbors and harder to produce recommendations for a particular user. In this work, we think that two users could be considered similar not only if they rated the same items similarly, but also if they belong to the same demographic segment. Accordingly, we do not look for neighbors who have a similar ratings pattern with the active user, but we directly categorize the user according his attribution. Pazzani propose a *Demographic filtering* which uses the demographic information, such as gender, age, area code, education, and employment information of users to learn the type of person that like a certain object in the restaurant recommendation application [26].

In this work, we choose item-based collaborative filtering to make recommendation. We analyzing user-item representations to identify relations between the different items, and then recommendations for a particular user are computed by finding items that are similar to other items the user has liked. Karypis [21] used item-based collaborative filtering for the top-N item recommendations. At Amazon.com the item-to-item collaborative filtering algorithm produces recommendations in real time, scales to massive data sets [23]. Moreover, Sarwar *et al*. [31] point out the bottleneck in collaborative filtering is that searching potential neighbors among a large user population. Since relationships between items are relatively static, item-based algorithms can avoid this bottleneck.

2.1.3 Hybrid Approaches

Collaborative Filtering methods collect items ratings from individuals and use nearest-neighbor techniques to make predict whether a user would be interested in a particular item. However, these methods miss the information about the nature of each item.Basu *et al*. [7] present a recommender approach that is able to use both ratings information and other forms of information about each artifact in predicting user preferences. Commonly, different ways to combine content and collaborative methods into a hybrid recommender system can be classified as follows:

1. To implement collaborative and content-based methods separately and combine their predictions.

2. To incorporate some content-based characteristics into a collaborative approach.

3. To incorporate some collaborative characteristics into a content-based approach.

4. To construct a general unifying model that incorporates both content-based and collaborative characteristics.

For example, Pazzani [26] proposed a *collaboration via content* approach which based on traditional collaborative techniques but also maintain the content-based profiles for each user. The content of the user's profile is not the commonly rated records, but contains weights for the terms that indicate that a user will like an object, and then is exploited to detect similarities among users. The benefit of this approach is that users can be recommended an item not only when this item is rated highly by users with the opinions of the like-minded users, but also directly against the user's profile.

2.2 User Model

Recommender systems are characterized by how to model users, and then connect similar ones together. As was observed, not many pairs of users will have a significant number of commonly rated items; the sparsity is a big problem in any rating-based recommender systems, therefore when model individuals, we cannot rely too heavily on having a large amount of rating for items or expecte users to answer many specific questions. To build user models quickly, some uncertain knowledge must be incorporated into modeling. Elaine Rich [29] uses stereotypes as a mechanism for building models of individual. Stereotypes are the clusters of characteristics of the user. With the aid of stereotypes, Grundy, a system that plays librarian, roughly models the readers, and then exploits those models to suggest relevant books that people may find interesting. The case in a library, the librarian knows some information about the reader before asking, such as the gender, approximate age, and some things he can assume until he has c ontrary evidences, like that the person whose hair is yellow does not read Chinese. Only when the librarian needs to figure out other more detail things, he asks the specific question to the reader. In other words, a user model was built without asking many questions to a user, but with making direct inferences from a user's behavior. Although not all of these attributes are necessarily true, the benefit of this approach is that user model is built quickly without making users feel tedious.

The classify algorithms can put the like-mined users into the same class, but each user only be clustered into a single cluster. In reality, a sound recommender system must be able to cluster users into several categories, for example, in a book recommendation case, a user may be interested in one topic (e.g., programming) for work purposes and a completely different topic (e.g., fishing) for leisure.

Moreover the user model is acquired not only based on user's charecter or his rating records, but also on the basis of user's activities in the information space.Oppermann and Specht practice a nomadic exhibition guide Hippie [25] for a museum guidance and introduce many psychology theories about modeling the user interests including, *motivation theories, psychological perception theories* ,and *social psychology*. According stereotypical movements of visitors, a visitor could be classified by four categories: ANT, GRASSHOPPER, FISH and BUTTERFLY. Given visiting style, the adaptive information guide can be present, long and detailed presentation for an ANT, short for a GRASSHOPPER and medium for a FISH and a BUTTERFLY. Konstan *et al*. [22] point out the correlation between spent reading time and explicit ratings. That is, the reader who spend a long time with an article is more likely to rate it highly; Similarly, other actions such as printing, saving, forwarding, and posting a follow up message to an article are also be a clue about how the user likes this article. There are some spot recommendation applications show correlation between user preference and travel behavior such as visit frequency and travel time [18][41][6]. For example, CityVoyager [41] models users' movements using first-order Markov model which uses areas as nodes. The transition probabilities of nodes are calculated from periodically plotted user locations, and a higher probability indicates more chances of a user advancing to the area. In this paper, we have no user's travel history at trip beginning, through analyzing his visiting behavior in the trip can help us to model the user. The approach chapter describes how to moale a user in detail.

2.3 Context-Aware Computing

The use of situational information is increasingly important in many fields such as ubiquitous computing where the context vary rapidly. Anind K. Dey [17] defines context：

> *Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves.*

He also defines context-aware：

> *A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task.*

Schmidt [36] defines Context awareness：

> *Context awareness as knowledge about the user's and IT device's state, including surroundings, situation, and to a lesser extent, location.*

Context-aware applications exploit the changes in environment and adapt according to user's location, collection of nearby people, and accessible devices, *etc.* In general, most context-aware applications focused on location awareness, location-based service (LBS) provides user information/service depending on the positon of entity. Here we introduce early work in context awarenes: Firstly, the Active Badge System [1] is a tag that periodically broadcasts a unique identifier for the purpose of determining the wearer's location, it was developed at Olivetti Research Lab. The main software

[1] source: `http://www.cl.cam.ac.uk/research/dtg/attarchive/ab.html`

application is an aid for a telephone receptionist. It redirect phone calls based on people's location real time [42]. Another work, the ParcTab [2] is a palm-size computer developed at the Xerox Palo Alto Research Center. It uses an infrared-based cellular network for communication wirelessly in an office setting. The system notifies applications of location changes, and also provides location information to a public service that collections and redistributes information about objects and their location. In [35], Schilit *et al.* use ParcTab to support four categories of context-aware applications: proximate selection, automatic contextual reconfiguration, contextual information and commands, and content-triggered actions.

In fact, context encompasses more than just the location of user, other contexts includes lighting, noise level, network connectivity, and even the social situation e.g., whether you are with your friend or with a family. Schmidt *et al.* organize context feature space into tow category, one is related to human factors, and the other is related to the physical environment. Human factors related context including information on the user, the user's social environment, and the user's tasks. Likewise, context related to physical environment including location, infrastructure, and physical conditions [37].

2.3.1 Context-Aware Technology in Tourist Domain

As usual these are two dimensions (user and item) in recommender system, all the remaining dimensions such as time, weather, and place will be called contextual dimensions since they identify the context in which recommendations are made. Steeen *et al.* [39] defined recommender systems as systems capable of finding what is interest

[2]source: `http://sandbox.xerox.com/parctab`

to a specific user through large amounts of information. Both context-awareness and recommender systems can enhance and complement each other, they both help users in finding relevant and interesting objects, the former based on the user's context, the latter based on the user's interests. Therefor, [40] Steeen *et al*. propose a open platform WASP which allows easy creation of context-aware personalized applications and the services.

In recent years, more and more related projects incorporate the contextual information into the recommendation in the tourism domain [14][15][28][5][40][9][2][41].

- **GUIDE** [15] is a intelligent and context-aware tourist guide. Cheverst *et al*. mention some issues and experiences of developing a context-aware tourist guide, such as dynamic information and context-sensitive information [14]. Then in [15] they focus on the presentation of adaptive hypermedia information within a intelligent and context-aware tourist guide, GUIDE. The context used in GUIDE includes the visitor's personal context (e.g. his current location) and the environmental context (e.g. the opening hour of attractions).

- **CRUMPET** [28] proposed services by taking advantage of integrating location-aware services, personalized user interaction, seamlessly accessible multi-media mobile communication, and smart component-based middleware technologies.

- **INTRIGUE** systme [5] is a prototype tourist information server; it provides an interactive agenda to assists user in scheduling an itinerary along tourist attractions based on location of each tourist attraction and user's interests. Specially, it recommends sightseeing spots tailored not only to individuals, but also to user groups, and explains the recommendations by addressing the group members' requirements. Before the INTRIGUE revising a itinerary, the visitor had to decide spots that he would like to see, what is start point, what is destination, and what kinds of transportation is prefer, *etc.* And during path planning, INTRIGUE makes some assumptions to simple questions, for example, the more spots in a trip the more fun the user has.

- **COMPASS** is a context-aware mobile personal assistant [40]. It is an application combine context-awareness and recommender systems that serves a tourist with information depends on the specific context that are interesting to him given his goal by a open platform WASP which allows easy creation of context-aware services.

2.3.2 Context-Aware Recommender System

Annie Chen[9] notices that traditional recommender algorithms have mostly been applied to applications for which the context is static, but the fact is that user's decisions are influenced by surrounding context; for that Chen proposes a context-aware collaborative filtering system that leverages the pervasive context information such that user's preference for an item is not only predicted from opinions of like-minded users, but also from feedback of other users in a context similar to that the user currently is in. How to measure similarities between contexts is the main issue, Chen devises a data driven approach that if the ratings for an item are similar for two different context values, then these two contexts are very relevant to each other. Given the context similarities and the current context surround the active user, the system give an overall prediction for the user on an item based on what others have chosen in a similar context in the past. If the amount of rating data is not enough, this data driven approach will not work well. In this case, we have to get the aid from domain knowledge. Similarly, Adomavicius *et al.* incorporate contextual information into recommendation to construct a multidimensional recommender system [2], and formulate the reduce process from multiple dimensions (user, item, time, and place) to the traditional two-dimensional (user, item) recommendation space. For instance, it is night o'clock in the Moring, a restaurant recommender system suggests customers what to eat based only on the data which occurred at 9:00 A.M. in the past; horever, in some cases, the database may not contain enough records at a specific time(9:00 A.M.) for two-dimensional recommender algorithm to predict accurately. When estimating an unknown rating, there is a tradeoff between having fewer but more relevant data and having more but less relevant data, for that Adomavicius *et al.* propose an algorithm to decide which contextual segment is the best for that particular rating.

2.4 Ontology

Ontology is adopted from philosophy where an ontology is a doctrine account of existence; in computer science field, ontology is an explicit and formal specification of a conceptualization composed by a finite set of terms and the relationships between these terms [20]. Moreover, ontology used as a way of specifying content-specific agreements for sharing and reuse of knowledge between humans and software entities; as a result, ontology has become common on the World-Wide Web, ontology includes machine-interpretable definitions of basic concepts and relations among them in a particular domain, for that electronic agents can understand and search for information, and makes semantic web vision come true [4]. Thomas R. Gruber proposes a set of criteria and principles to guide the development of ontology [20], and Noy and McGuinness use a wine and food example to explain why and how to creating an ontology using Protégé [3] step by step [24].

In recent years, many famous projects have been designed ontology used for knowledge sharing.

- **EasyMeeting** is an intelligent meeting environment [11], it using a context-aware pervasive computing framework called CoBra (COntext BRoker Architecture) [10] to model the basic concepts of places, agents, events, and their associated properties in meeting domain to provide knowledge sharing, context reasoning and privacy protection.

- **SOUPA** is the acronym of Standard Ontology for Ubiquitous and Pervasive Applications [13][12].It intended to be a standard specification of ontology technology, it includes modular component vocabularies to represent intelligent agents with associated beliefs, desire, and intentions, time , space, events, user profiles, actions, and policies for security and privacy.

- **MyCampus** [30] developed at Carnegie Mellon University. they aim at enhancing everyday life and provid a Semantic Web infrastructure for Ambient Intelli-

[3] source: `http://protege.stanford.edu/`

gent. Whitin this infrastructure, each contextual information (e.g. a calendar, location tracking functionality, user preferences) is represented as a service. Also they implemented campus activity ontology for faculty and students to infer their possible location and available situation.

In short, ontology is a commonly approach of specifying data semantics. [44] using ontology to model geometric relations between buildings in National Taiwan University (NTU)campus and using Semantic Web Rule Language (SWRL) to infer new knowledge such as building A connects with building B and building B is connects with building C then building A is indirect connects with building C. In this thesis, a geographical ontology is built to organize spot information and to conceptualize the campus geographic knowledge, such as hierarchical structure, included-in, nearby relations [19][27][43]. In addition, we using an upper ontology [4] of temporal conceptions, and adding temporal individuals and interval according the campus guidance knowledge, for example the open hours of the main library.

[4]source: `http://www.isi.edu/`

Chapter 3

Context-Aware Campus Scenic Spots Recommender

In this thesis, we implement a context-aware campus spots recommender system which actively suggests the visitor adjusting his trip depend on the landmark properties, visitor preferences and contextual information includes current time, weather and company. In this chapter we formulate this context-aware recommender problem and identify the challenges, then propose and explain our solution in detail step by step. First, we undertake an activity to collect spots ratings with different context and visitors' moving patterms over NTU campus. Analyzing those data, item-based collaborative filtering is used to compute the similarity among spots in NTU campus; moreover, a user oriented location profile was build to enhance the spatial model built with ontology.Next, through the contexts filtering and relaxing processes, we compute the high performance contextual-segments based on the collected dataset. Finally, dependents on current context a significant contextual-segment can be determine, then we assign a utility to each not yet rated spots to create a recommendation based on user's responses of what has visited in trip and attractions information.

3.1 Problem Formulation

Intuitively, the recommendation problem is reduced to the problem of rating estimation for items that have not been seen by a particular use, in other words, the main task in recommender systems is to predict the votes of a particular user over given subjects. There are many different ways can be use to estimate the ratings such as machine learning, approximation theory, and various heuristics, *etc.* In this work, we incorporate contexts into recommend process, and use item-based collaborative filtering to figure out correlation between spots according to currrent time and weather, then use category-base prediction on user side to make the rating prediction. Once ratings of the not yet rated spots have been estimated, a designed utility function is used to produce a list of landmarks with the higher scores for the particular user.

There are 110 buildings, more than 15 hotspots at NTU Campus including Royal Palm Boulevard, Fu Bell, Main Library, Siao Fu Commissary and Liugongjun Pool, *etc.* Since in the restricted geographical area, the number of spots is static and the properties of spots are diversified, we chose landmarks in NTU campus as our recommended items to implement the context-aware recommender system.

Formally, the context-aware recommendation problem is formulated as follows: Let POI is the set of spots that can be recommended. In the rest of this thesis, we call each point in POI as Points Of Interest (POIs), each element in POI is defined with a set of characteristics, a spatial ontology is used to model it. Let U is a utility function that measures the degress of interested in each POIs to a particular user, and the utility factor including, visitor's characteristics, spots profile, and environmental conditions. Then, for each user, the goal of the recommender system is to select a set of items $POI\prime \in POI$ which has the higher utility. All criterions in the utility function will be seen clearly in the last part of this chapter.

3.2 Challenge and Proposed Solution

In this work, the system needs context-aware abilities, a thorough-considered campus knowledge base, a flexible recommender algorithm, and an efficient user modeling. In view of the preceding purpose, three major sets of research questions to be addressed on this study are as follows:

- **User oriented location profile :** Spots are not text-based recommended items causes that the information of spots cannot be gotten through content-based analysis techniques. Here we propose a user oriented location profile of which spots characteristics are extracting from users' visiting behaviors over NTU campus. On the one hand, the rates for spots are used to compute the similarity among spots by adjusted correlation-based approach, on the other hand, the moving patterns are used to find some latent relationships among locations, for instance, in our collected dataset, most people go to full bell then go to main library, we define the *nextStop* relation between full bell and main library.

- **Context-aware recommendation:** There is a tradeoff between having fewer but more relevant data and having more but less relevant data for estimating an unknownn rating. According to the campus spots recommendation application we have to determine which contextual varialbes are important,and then a contextual-segment determine algorithm is introduced to decide what degree data with the contextual information should be incorporated into the recommendation process.

- **New user:** We want to give recommendations to new user whose preference cannot be retrieved at the trip beginning, in order to make more accurate recommendations, the system get familiar with the user via not only explicit ratings given by this user for items, but also observation the user behavior in the trip incrementally. In other words, the user profile is both explicitly provided by the user and is implicitly inferred by the system. Besides, since the new user problem, the user-based collaborative filtering recommender algorithms are not appropriate for this work.

Accoring to the above challenges we propose a solution in Figure 3.1.

3.2.1 Data Collection and Analysis

As general rating based recommender systems, a large number of users' rate for the items is required for maiking prediction. We held an activity on NTU Azalea Festival to collect spots ratings with context and visitors' moving patterns over NTU campus. The participants visit NTU campus along with a GPS logger which can store their moving

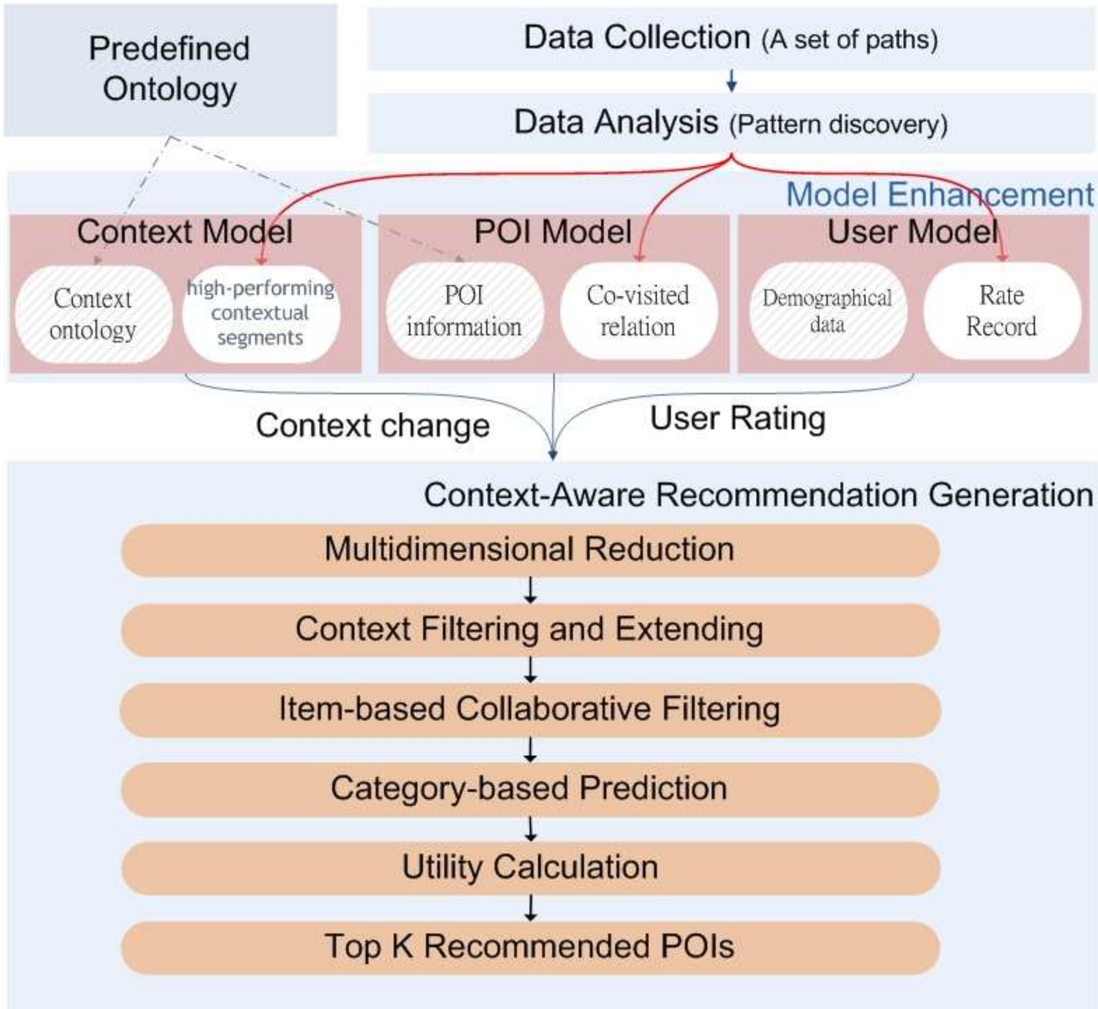

Figure 3.1: Solution Flow Chart

tracks. We also design a questionnaire to record the visitors' features such as gender, age, and who in the same group(single, couple, family, friend, *etc.*). Meanwhile, we required people to grade landmarks in the trip base on what time is it, whether it rain , and companions are who. The rating scale is from 1 to 5, the higher marks the more interested.

We have collected 93 logs, 93 user profiles, and 949 user explicit ratings for POI associated with context. Let D is logs dataset, a log is composed by a user profile and several POI rated records. Each POI rated record includes the following fields: POIid, rate, rateTime, exitTime, like, weather,air temperature, visitedTimes, photo, note, indoor.

$$D = \{log_1, log_2, \ldots, log_{93}\}$$

$$log_u = (userProfile_u, POIRatedRecord_{u,1} \ldots, POIRatedRecord_{u,k})$$

$$\begin{aligned}
POIRateRecord_{u,t} &= (POIid_{u,t}, explicitRate_{u,t}, rateTime_{u,t}, exitTime_{u,t}, like_{u,t}, \\
&\quad weather_{u,t}, air_{u,t}, temperature_{u,t}, visitedTimes_{u,t}, photo_{u,t}, indoor_{u,t}) \\
POIid_{u,t} &\in \{1, 2, \ldots, 143\} \\
explicitRate_{u,t} &\in \{1-5\} \\
rateTime_{u,t} &\in \{00:00-23:59\} \\
exitTime_{u,t} &\in \{00:00-23:59\} \\
like_{u,t} &\in \{yes, no\} \\
weather_{u,t} &\in \{\text{clear, cloudy, rainy, null}\} \\
air_{u,t} &\in \{\text{calm, breeze, blustery, null}\} \\
temperature_{u,t} &\in \{\text{cold, cool, warm, hot, scorchingheat, null}\} \\
visitedTime_{u,t} &\in \{1, 2, \ldots\} \\
photo_{u,t} &\in \{0, 1, 2, \ldots\} \\
indoor_{u,t} &\in \{yes, no\}
\end{aligned}$$

We incorporate contextual information into recommend process, in other words, the recommendation space is not only two dimensions (User, Item) but it is multiple dimensions, such as User, Item, Time, Weather. When defining the multiple dimensions recommendation space, it is important to understand what dimension should be included according to the applications. For movie recommender system apart from user and movie(item), where to see the movie affects user's decision and whether it rains is inconsequential, but for scenic spots recommender system the weather become a important consideration.

Considering the application that recommends scenic spots to the visitors, first we use domain knowledge to choice dimensions, then we test which dimension really matter with respect to making a significant difference in rating estimations by analysis of variance (ANOVA) and analysis of covariance (ANCOVA). In collected dataset D, the temperature is personal feeling and it is too subjective to use. The air volume information is incomplete, therefore we exclude those two factors. Then, we apply one way ANOVA to see the effect the remaining factors. Explicit rate is dependent variable, the others, such as POIid, weather, hours of day, age, tourist group, and gender are independent variables respectively, and $p < 0.05$ was defined as statistically significant. The results of Table 3.1 show that the POIid, wather, age and tourist group explained of the variance of rate; especially, both POIid and age have a statistically significant impact on rate.

If there is a interaction between factors, the changes of a factor cause the variance of rate would be affected by the other factor. Two way ANOVA is used to check whether the interaction between two factors is significant (at the 0.05 level). Table 3.2 shows that the interaction between hours of day and POIid has a significant impact

Table 3.1: One-way ANOVA

Test of between-subject effects (one-way ANOVA)			
Dependent Variable : explicit rate			
POIid	F(80,864)=2.158	p=.000	$p < .001$
weather	F(3,941)= 3.045	p=.028	$p < .05$
hours of day	F(8,936)=1.894	p=.058	
age	F(34,910)=2.271	p=.000	$p < .001$
toursit group	F(6,938)=3.148	p=.005	$p < .01$
gender	F(1,943)=2.896	p=.089	

Table 3.2: Two-way ANOVA

Test of between-subject effects (two-way ANOVA)			
Dependent Variable : rate			
POIid $\times$ weather	F(90,771)= 1.110	P =.238	Interaction is not significant
POIid $\times$ hours of day	F(196,660)= 1.565	P =.000	$p< .001$
POIid $\times$ age	F(403,427)= 1.106	P =.152	Interaction is not significant
POIid $\times$ toursit group	F(144,714)= 1.009	P =.460	Interaction is not significant
POIid $\times$ gender	F(39,824)= 1.511	P =.025	$p< .05$

Table 3.3: ANCOVA

analysis of covariance (ANCOVA)			
Dependent Variable : rate			
Covariate : POIid			
Weather	F(3,940)= 2.839	P =.037	p<0.5
Age	F(34,909)= 2.273	P =.000	p<0.001
Toursit Group	F(6,937)= 3.261	P =.004	p<0.01

on rate, it meant that the time influence on rate affected by different POIs. Similarly, the interaction between gender and POIid has a impact on rate, it meant that male and female like different POIs. As a consequence, for each POI we compute the popular hour of day and proper gender to enhance POI model and to improve recommendations quality. Table 3.3 shows that according analysis of covariance which covariate is POIid, regardless of the change in POI, the wather, age and tourist group explained of the variance of rate.

To sum up, after the rating data had been analyzed, we decide the following dimensions:

$$\text{Recommendation Space} = User \times POI \times Time \times Weather \tag{3.1}$$

$$\begin{aligned} User &\subseteq \text{ gender} \times \text{age} \times \text{companion} \\ POI &\subseteq \text{ POIid} \times \text{genery} \\ Time &\subseteq \text{ hours of day} \\ Weather &\in \text{ clear,cloudy,rainy} \end{aligned}$$

Each dimension has some attributes to define it, for example, the $User$ dimension represents people for whom POIs are recommended in this application, it is described by age, gender and what kind of group he belongs in. Similarly, the POI dimension is a set of POI having certain identification number. The $Time$ dimension describes what hours in a day when the POI is visited. Finally, the $Weather$ dimension represents the weather conditions (clear, cloudy, or rainy) when the POI is visited. Given recommendation space, the rating prediction function R specifies how much user $u \in User$ liked POI $p \in POI$ at time $t \in Time$ and $w \in Weather$.

$$R^D(u, p, t, w) :\rightarrow \text{rating} \tag{3.2}$$

3.2.2 POI Model

We build a spatial ontology $POIModel$ to store the attributes of scenic spots in NTU campus. Let POI_i is a POI profile i.e., a set of attributes characterizing POI i. Each POI profile is defined with a identical number $POIid_i$, a name $title_i$, the geographical position is denoted by latitude lat_i and longitude lon_i, the category $catgory_i$ it belongs to, the spot genres $genres_i$, the available time form $openTime_i$ to $closeTime_i$, and

in_i presents whether the POI is a indoor spot.

$$POIModel = \{POI_1, POI_2, \ldots, POI_n\}$$

$$POI_i = (POIid_i, title_i, lat_i, lon_i, category_i, genres_i, openTime_i, closeTime_i, in_i)$$

$$POIid_i \in \{1, 2, \ldots, 143\}$$

$$catgory_i \in \{\text{Academic,Administrative,Dormitory,Instructional,LifeRecreateion,Hotspot}\}$$

$$genres_i \in \{\text{Architecture,Ecology,Museum,Academy,Entertainment,Workout,Hotspot}\}$$

$$openTime_i \in \{00:00 - 23:59\}$$

$$closeTime_i \in \{00:00 - 23:59\}$$

$$in_i \in \{yes, no\}$$

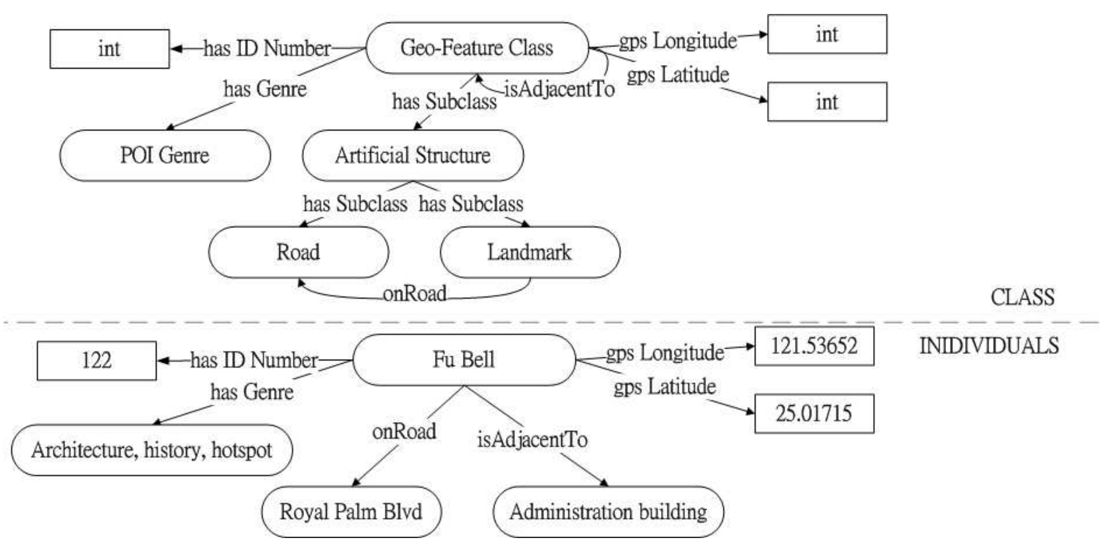

Figure 3.2: Spatial Ontology

The POI category defined within the location ontology, mainly based on the functionality a POI provides. The main category includes academic building, administrative unit, dormitory, instructional building, life-recreateion, and hotspot. Some categories have subcategories such as under academic building including college of bio-resources, college of engineering and college of management. The definition and hierarchy structure of the category can be re-assigned if needed for new data. In addition, the spatial ontology defines a set of spatial relatinship among spots, for example Fu Bell and Administration building are adjacent, and the Fu Bell on road of Royal Palm Blvd.. Part of location model in the representation of ontology is shown in Figure 3.2.

Analyzing the primitive data is useful for capturing some evolving behaviors to advance POI profiles. For each POI, apart from the basic attributes of attractions which have been defined within the ontology, there are some features learned from the visitors' behaviors, including the popularity $popularity_i$, average duration of stay(in minutes) $avgStay_i$, when is the proper time $properTime_i$ to visit for POI i, and what gender people $properGender_i$ is more like POI i.

$$
\begin{aligned}
POIModel_{rich} &= \{POI_1 + enrichInfo_1, POI_2 + enrichInfo_2, \ldots, POI_n + enrichInfo_n\} \\
enrichInfo_i &= (popularity_i, avgStay_i, properTime_i, properTimeMean_i, \\
&\quad properGender_i, properGenderMean_i) \\
popularity_i &\in \{1 - 5\} \\
avgStay_i &\in \{0, 1, 2, \ldots, 1240\} \\
properTime_i &\in \{00:00 - 23:59\} \\
properTimeMean_i &\in \{1 - 5\} \\
properGender_i &\in \{\text{male, female}\} \\
properGenderMean_i &\in \{1 - 5\}
\end{aligned}
$$

Those advance information $enrichInfo$ can provides clues to recommendation. For example, there is a path pattern, most people prefers to visit library at 11:00 AM , i.e. POI = library →popular HourOfDay =11:00 AM.

Additionally, for the relation among the POIs, apart from the geographic relations (e.g., distance, nearby, overlap...) among attractions which have been defined within ontology, the similarities between any two POIs are computed by analyzing the ratings, and the link intensities among POIs are calculated by analyzing the order of visited POI in each paths. In short, those information was used to strengthen the original POI model, and the enhanced POIModel $POIModel_{rich}$ is used to determine the appropriateness of the spot for recommendation purposes.

3.2.3 User Model

We want to know whether the user like the spot we recommended, we have to understand what he like and what he does not like. The interests of individual are the most difficult part to model, especially because they are highly dynamic. Most of the state-of-the-art recommender services use collaborative filtering algorithm to recommend items to a particular user based on the opinions of other like-minded people. However, a natural limitation of CF is that while CF makes a high quality recommendation, it needs to have a large scale of user-item ratings. Particularly, in this work we assume that the user is a new one who has no preference on any POIs and no user of the community. Since we think that two users could be considered similar not only if they rated the same items similarly, but also if they belong to the same demographic segment. In this work, rather than computing user similarities by user-item ratings, we directly exploit the opinions from who is in the same age level or plays with the same tourist group to complement the sparsity problem.

This paper proposes two phase method to acquire knowledge about users, we ask the user some basic information(stereotype) to build a rough user model quickly, and then during the trip we record how he satisfied with current arrangement and his visiting behavior to implicitly predict his preference. In particular, the system records the user activities whenever he visits a spot, browses a spot introduction, or receives a suggestion and tries to infer the user's actual preferences. In other words, the subsequent user interaction with the system during the trip would enhance the initial profile. For example, we observe the user's rate distribution in term of POI genres to infer what kind of POIs he like more dynamically. Since we think that a person may interested in different types of attractions depend on the moment, the weather, and the company. For instance, a man may be interested in topic (e.g., historical) when his friend company with him and a completely different topic (e.g., academic) when he come with his children. In short, the user's preference is empty in the begin, as time goes by, it refines by user's visiting and usage behavior. The benefit of this approach is that the user's profile is dynamic and constantly update without disturbing the visitor.

phase one: stereotype modeling

The principle of modeling user is not disturb the user, therefore asking the more precise question the better. In the initial stage we ask a user only a few precise questions which allow user be segmented along demographic information e.g., age, gender, and who come with him in this trip. Formally, we build a user model $UserModel$ to store the attributes of visitors, everyone can be characterized with a user profile $userProfile$.

$$
\begin{aligned}
UserModel &= (userProfile_1, userProfile_2, \ldots, userProfile_m) \\
userProfile_u &= (userId_u, gender_u, age_u, touristGroup_u) \\
userId_u &\in \{1, 2, \ldots\ m\} \\
gender_u &\in \{\text{male, female}\} \\
age_u &\in \{1, 2, \ldots, 100\} \\
touristGroup_u &\in \{\text{alone, couple, faimily, family with child, family with elder, friend, other}\}
\end{aligned}
$$

phase two: explicit and implicit ratings

In the second stage, during a trip the system require users to rate for POIs explicitly. We cannot expect users involve in user modeling process actively, in reality, few people have rated most of item, especially when taking contextual information into recommend process, the sparsity problem become worse. Therefore, we must be able to exploit at best indirect knowledge about the user's behavior[22][16]. We observating users' visiting and usage behaviors to infer a implicit rates for POIs. In short, the knowledge model of the user can be refine by the explicit ratings given by this user and monitoring the user's interaction with mobile device to learn the implicit ratings over time.

$$
\begin{aligned}
UserRatingRecord_{u,t} = (&userProfile_u, POIid_{u,t}, explicitRating_{u,t}, implicityRating_{u,t},\\
&ratingTime_{u,t}, exitTime_{u,t}, ratingWeather_{u,t})\\
POIid_{u,t} &\in \{0, 1, 2, \ldots, 143\}\\
explicitRating_{u,t} &\in \{1-5\}\\
implicityRating_{u,t} &\in \{1-5\}\\
ratingTime_{u,t} &\in \{00:00-23:59\}\\
exitTime_{u,t} &\in \{00:00-23:59\}\\
ratingWeather_{u,t} &\in \{\text{clear, cloudy, rainy}\}\\
photo_{u,t} &\in \{0, 1, 2, \ldots\}
\end{aligned}
$$

Again, in this work the user profile is built not only based on the explicitly ratings $explicitRating_{i,t}$ assigned by user u for the t-th POI he rated in the trip, but also implicitly rating $implicityRating_{u,t}$ inferred by the system. The rest varialbes $POIid_{u,t}$ is detectd by GPS logger, $ratingTime_{u,t}$ and $ratingWeather_{u,t}$ are recorded automaticly when user u rate for POI.

The explicit voting refers to a user consciously expressing user preference for a spot, and the implicit voting refers to interpreting user behaviors, such as the time spent on each visited POI, the number of photo he take, or the browsing pattern. Through statistical significance analysis, if a user has requested significantly more than the average amount of a certain kind of information, or has token more time than average on a certain item, he is probably interested in this kind of information. We make assumptions that the longer the user stays at the POI, the more interest at it, and the more pictures user take for the POI, the more interest at it; the former required location-specific normalization, so we compare his stays time with the average one, and the latter required personalized normalization. In addition, for most POI of NTU campus there is an introduciton webpage, visitor can browse the content via our system, meanwhile the time spent on pages of the user is computed to model user preference. Similarly, we think that the reader who spends longer time on a page is more likely to rate it highly. Apart from the reading time reflects user interests, other user activities is also useful, including review the POI, accept or ignore the recommend POI. Positive evidence for gaining knowledge about objects comes from reviewing or adding POIs. Negative evidence comes from skipping or ignoring POIs.

From what is said above, each POI rates takes two part: explicitly and implicitly.

The user denoted as u , the POI denoted as p, and $r_{u,p}$ is the grade of POI p rated by user u .

$$r_{u,p} = w_{ex} \times EXr_{u,p} + w_{im} \times IMr_{u,p} \tag{3.3}$$

where w_{ex} is weight of explicit rating and $0 \leq w_{ex} \leq 1$, w_{im} is weight of implictiy rating and $0 \leq w_{im} \leq 1$, and $w_{ex} + w_{im} = 1$.

The implicitly rating plays a critical role in context-aware recommender system, because it is too verbose if we ask the user grade the same POI every ten minutes.

3.2.4 Context Model

The input of this recommender system involves the ehrich POI model, User model, and environmental context. The primitive contexts involves the user's position, current time, and the weather condition: the reference position (ϕ_t, λ_t) is the latitude and longitude to specify the user's corrent location, the instant time in calendar clock is characterized by the hour-of-day $hour_t$, and the weather condition denoted as $weather_t$ to say whether it rains.

$$Context_t = (\phi_t, \lambda_t, time_t, weather_t)$$

$$hour_t \in \{0, 1, 2, \ldots, 23\}$$

$$weather_t \in \{\text{clear, cloudy, rainy}\}$$

This context-aware spots recommender system is operated outside, therefore a Global Positioning System (GPS) receiver is used to detect the visitor position. On the one hand, each POI on NTU campus were recorded its absolute coordinates location in spatial ontology. Also, some spatial relationship among spots are defined in this spatial ontology (see Figure 3.2).

On the other hand, we build a temporal ontology to model time knowledge. First, we divide the temporal entity into two main categories instant and interval, each interval has a start instant and an end instant, and each instant is defined with temporal unit (year, month, day, hour, minute, and second). Second, some high level temporal concepts are described in this temporal ontology, such as, a day includes several intervals: morning, noon, afternoon, evering, *etc.* Last, according to the application that ontology apply, we define some temporal concepts in tourist domain. For example, open-hour and close-hour of main Library tell the system what time is available for Library. Figure 3.3 shows some temporal relationships: A weekday *contains* Monday to Friday and each day contains some intervals: morning, noon, afternoon, evening, night, *etc.* The *before* relationship among Sunday, Monday and Tuesday helps the computer know that the day after Sunday and before Tuesday is Monday. And the *begins* and *ends* properties specify the library open at 8:00am and close at 10:30pm on Monday. Therefore this temporal ontology can be used to infer which landmark is suitable to recommend according what time it is now.

The weather-aware function detects whether it is raining and infers what spot is not suitable to recommend. Weather information is retrieved from the web site of department of Atmospheric Sciences, NTU [1]. It provides the information of temperature, humidity, cumulative rainfall in one minute, cumulative rainfall in one day, and atmo-

[1] source: `http://www.as.ntu.edu.tw/`

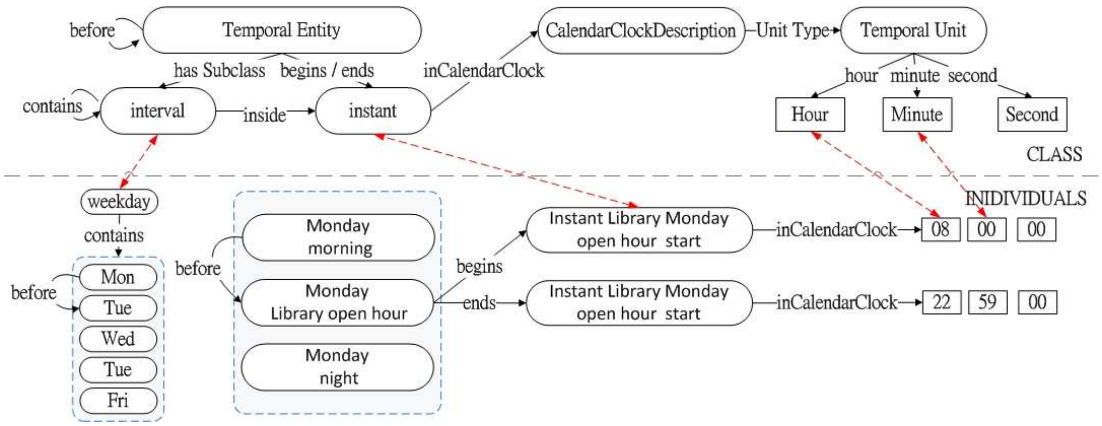

Figure 3.3: Tempral Ontology

spheric pressure. It updates frequency is one minute. In this work, we use temperature, humidity and cumulative rainfall in one minute to define current weather condition.

3.2.5 Context-Aware Recommendation Generation

To understand when and how to propose a recommendation to user, consider a case when it rains suddenly, the system detects the changes in the environment, and takes user's position, POI information, and current contextual information (time and weather) as reference, then context-aware recommender process begins. (see Figure 3.4)

stage one: multidimensional reduction

We would like to reiterate that all user-specified ratings in this work was associated with time and weather, our recommendation space is four dimensions which was defined in Euqation (3.1). How to estimate the unknownn ratings in multiple dimensions recommendation space. The first step is to reduce the multiple dimensions to two

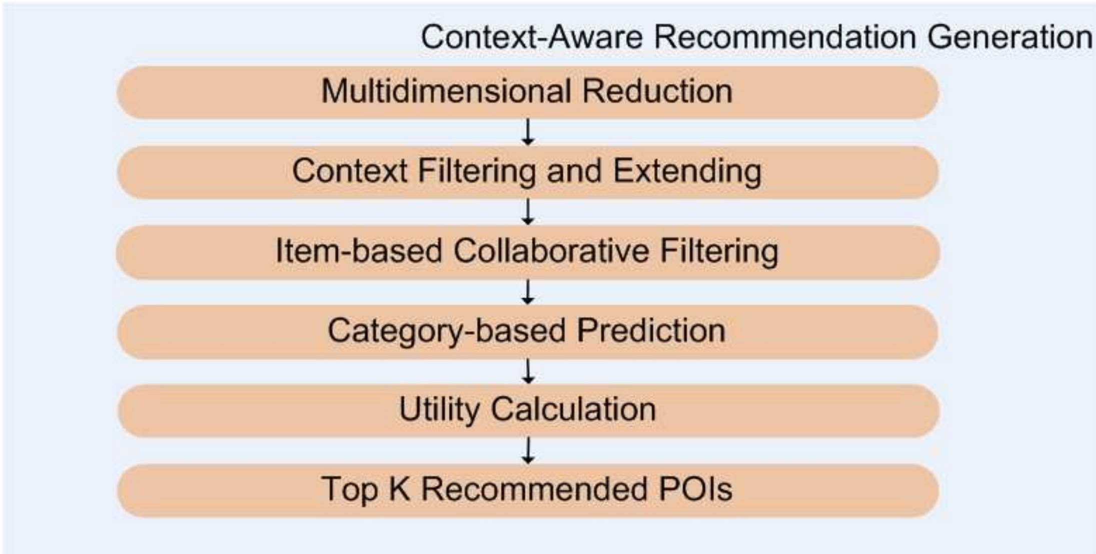

Figure 3.4: Context-Aware Recommendation Generation

dimensional(User and Item), then the traditional recommender algorithms such as collaborative filtering can be applied directly. Precisely, dimension reduction is actually to remove the data of which context is different from current context. Consider a case it is 9:00(t) and it is raing(w), the reduce step is to retrievie the data which was occured at 9:00 and rain in the past.

$$\forall (u, p, t, w) \in \text{User} \times \text{POI} \times \text{Time} \times \text{Weather},$$
$$R^D(u, p, t, w) = R^{D[t=t\prime, w=w\prime]}(u, p)$$

where $D[Time = t\prime, Weather = w\prime]$ denotes a rating set where $Time$ dimension is value $t\prime$ and $Weather$ dimension is value $w\prime$, that meant the records with exactly the same contexts will be pickup from D. Hovever, in some cases the strongly context filtering causes data sparse, the rating dataset $D[Time = t\prime, Weather = w\prime]$ may not contain enough ratings for two dimensional recommender algorithms to predict unknown ratings accurately. Therefore, if the restricted dataset is not large enough, the second step is to relax the context constraint.

stage two: context filtering and extending

The prediction function $R(u, p, t\prime, w\prime)$ not only refers to data which exactly occurred with the specific context(t,w), but refers to a contextual segment $S_{t\prime, w\prime}$, which denotes some superset of the context $t\prime, w\prime$, for instance, (9:00,rainy) can be relax to its superset 9:00. This poses another problem: which level of contextual segment should be extend based on current context. We need to know which contextual segment is the best for making rating according to the particular context. Adomavicius *et al.*[2] propose a algorithm for determining high-performing contextual segments. We modify and implement it to produce meaningful contextual segments in term of rating prediction according to our dataset.

Here, **mean absolute error(MAE)** is used as performance metric μ, which compares the predicted ratings against the actual user ratings on the test set, the lower MAE, the more accurately the rating prediction.

$$\mu_X(Y) = (1/|Y|) \sum_{p \in Y} |R_X(u, p) - r_{u,p}| \tag{3.4}$$

where $\mu_X(Y)$ is the performance metric for our recommendation algorithm trained on the set of known ratings X (training set) and evaluated on the set of known ratings Y (testing set), where X $\bigcap$ Y = $\oslash$. As mentioned above, we have collected 93 visitor logs, we use leave-one-out cross-validation to form the training and testing set. This means that the recommendation algorithm tests every log by using the other 92 logs as trains data respectively. For each point $p \in Y$, $r_{u,p}$ is the user-specified rating for POI p , and $R_X(u,p)$ is the predicted rating on POI p for User u trained on dataset X.

We describe how to produce the high-performing contextual segments in Algortihm 1. First step, all contextual segments are produced by *Cartesian product* from each

Algorithm 1 Contextual Segments Produceing

inputs: T, set of user-specified ratings associated with time and weather.
$\mu_X(Y)$, performance metric based on training set X ,testing set Y
$Time$, Time dimension
$Weather$, Weather dimension

outputs: $SEG(T)$, set of contextual segments on which the context-aware recommender algorithm A do well.

$SEG(T) \leftarrow \{\}$
$AllSeg(T) \leftarrow \{\}$
for all t_i in $Time$ **do**
  for all w_i in $Weather$ **do**
    $AllSeg(T) \leftarrow (t_i, w_i)$
  end for
end for
for all S_i in $AllSeg(T)$ **do**
  if $\mu_{S_i}(S_i) < \mu_T(S_i)$ **then**
    $Q \leftarrow S_i \subset Q$
    if $\mu_{S_i}(S_i) < \mu_Q(Q)$ **then**
      $SEG(T) \leftarrow SEG(T) \cup S_i$
    end if
  end if
end for
return $SEG(T)$

elements in $Time$ and $Weather$ dimensions. Second step, for each contextual segment S_i , we train recommendation algorithm on the training data S_i to evaluate test set S_i, then compute its performance $\mu_{S_i}(S_i)$. Meanwhile we run algorithm on the whole data T to evaluate test set S_i, then compute the performance$\mu_T(S_i)$. We comapre those two performance if $\mu_{S_i}(S_i)$ outperforms $\mu_T(S_i)$, moving to step three. If there is a more general segment Q of segment S_i, we compare $\mu_{S_i}(S_i)$ and $\mu_Q(Q)$, and reomve the contextual segment S_i where $\mu_Q(Q)$ performs better than $\mu_{S_i}(S_i)$ and store $\mu_Q(Q)$. In addition, time ontology rally helps us to extent current time to high level temporal concepts. We use ontology to define uncontinuous temporal concept(e.g., eating time includes 9:00, 11:00, 12:00, 13:00, and 17:00), and then the specific context constraint can be relaxed to a super-concept according to the hierarchy relations in temperoal ontology, for example, (9:00,clear) can relax to (eating time,clear).

Once the set of high-performance contextual segmets $SEG(T)$ is computed, depends on current time and weather we can get a significant contextual segment SIG_{S_i} ,and use it as the train data for recommender algorithm. In this scheme, a specific context (t, w) relaxes to it's super-concept gradually, briefly, in each relaxation step, one context dimension is picked and the value is relaxed to one of its immediate super-concept, for example, (9:00, clear) can be relaxed to (9:00, all weather) which relaxes the weather dimenstion from clear to all weather. After context relaxation, more relevant records which were rated under the context similar to or exactly the same with the current context with respect to rating estimation will be picked out.

stage three: item-based collaborative filtering

Next we step into the recommender process, item-based collaborative filtering is used to computing POI similarities. Since we assume that user's preference cannot be learned at the trip beginning, traditional user-based collaborative

filtering will fail to find neighbors. Rather than matching the visitor to the like-minded people, item-based collaborative

filtering matches each of the user's rated items to similar items, it compute POI similarity by analyzing user-item representation in the significant contextual segment. More precisely, **adjusted cosine similarity** was applied to measure the correlation between two different POI with respect user-item representation, and it is a general approach to find the degree of correlation among items, rather than relying on just the most similar item.

In item-based CF, each items pair in the co-planned set corresponds to a different logs, then taking the differences in rating scale among different users into account, the adjusted cosine similarity subtracts the corresponding user average from each co-rated pair.

$$sim(p_i, p_j) = \begin{cases} \frac{\sum_{u \in co_{p_i,p_j}} (r_{u,p_i} - \overline{r_u})(r_{u,p_j} - \overline{r_u})}{\sqrt{\sum_{u \in co_{p_i,p_j}} (r_{u,p_i} - \overline{r_u})^2} \sqrt{\sum_{u \in co_{p_i,p_j}} (r_{u,p_j} - \overline{r_u})^2}} & \text{if } co_{p_i,p_j} \text{ is not empty} \\ 1 & \text{if } p_i \text{ and } p_j \text{ are the same POI} \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

where similarity $sim(p_i, p_j)$ reflects the degree of correlation between POI p_i and p_j, it is a continuous value from -1 to +1, the bigger value means the more connection between POI p_i and p_j. There is an observed tendency that when users who visited POI p_i also visited POI p_j and rated them in similar scale, the similarity $sim(p_i, p_j)$ increases. The user opinions for POI p_i of in log u denote as r_{u,p_i}, it consists of explicit part given by the user, and implicit part infered from user behaviors, the w_{ex} and w_{im} are the weight of explicit and implicit preference respectively. Rating mean of log u

is denoted as $\overline{r_u}$. Furthermore, co_{p_i,p_j} denotes a set of logs which both include POI p_i and POI p_j, we call it as the co-planned cases. Algorithm 2 shows how to find the co-planned case for each POI pair.

Algorithm 2 co-planned pathlog forming

inputs: D, the collected pathlogs which includes User-POI representations
POI, POI dimension
outputs: co_{p_i,p_j}, a set of pathlogs which both include POI p_i and p_j,

$co_{p_i,p_j} = \oslash$
for all p_i in POI **do**
  for all p_j in POI **do**
    for all log_u in D **do**
      if p_i rated in log_u and p_j rated in log_u **then**
        return $co_{p_i,p_j} = co_{p_i,p_j} \bigcup log_u$
      end if
    end for
    return co_{p_i,p_j}
  end for
end for

stage four: category-based prediction

After analyzing User-POI representations to identify the similarity between any two POIs, the next step is to predict the unknown ratings for a particular user. Accurately, the vote prediction $R(u,p)$ of user u for POI p is weighted sum of votes for the POIs given by the active user and users similar to him.

$$R(u,p_i) = \frac{\sum_{j=1}^{n} sim(p_i,p_j)(w_u \times r_{u,p_j} + w_{su} \times \overline{r}_{su,p_j})}{\sum_{j=1}^{n} |sim(p_i,p_j)|(w_u + w_{su})} \tag{3.6}$$

Each rating r_{u,p_j} is weighted by the corresponding similarity $sim(p_i,p_j)$ between POIs p_i and p_j. The w_u is the weight of active user, w_{su} is the weight of users whose age is in the same age level with that of active user, or whose companion is the same with that of active user. We would like to reiterate that the similar users in this word are defined as who belongs to the same age level or tourist group with the active user, instead of computing users similarity by analyzing users' specific ratings. We defined a similar users set of user u SU_u, it includes users who is in the same age level with active user u, or whose companion is the same with that of active user. The $\overline{r}_{su,p_j}$ is the average ratings for POI p_j of all similar users, it was computed by Equation (3.7), (3.8).

$$\overline{r}_{su,p_j} = \frac{\sum_{\acute{u} \in SU_u} r_{\acute{u},p_j} \times \delta(\acute{u}, p_j)}{\sum_{\acute{u} \in SU_u} \delta(\acute{u}, p_j)} \tag{3.7}$$

$$\delta(\acute{u}, p_j) = \begin{cases} 1 & \text{if } \acute{u} \text{ had rated POI } p_j \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

stage five: utility function

Until now, we know how the system estimates the unknown ratings. In the last step, we use a utility function U whihc takes user-oriented POI attributes into account to assign each POI a degree of hotness, and then actual recommendations of POIs to a user are made by selecting the k hottest POIs.

$$
\begin{aligned}
U(u, p_i) &= \sum_i a_i u_i \\
u_1 &= R(u, p_i) \\
u_2 &= popularity(p_i) \\
u_3 &= inProperTime(p_i) \times ProperTimeMean(p_i) \\
u_4 &= isProperGender(p_i) \times ProperGenderMean(p_i) \\
u_5 &= \sum_{p_j \in V} r_{u,p_j} \times EqGenre(p_i, p_j)
\end{aligned}
$$

where V is the set of POI which had rated by User u, and for each POI which has not be visited, we assign it a hotness not only by estimating its rating based on user-POI ratings, but we also consider its popularity, popular time of day, and the gender of the visitor. In addition, we design a funtion $eqGenre$ to compare the genres of each not visited POIs with POIs which has rated by the user, the idea behind this definition is that user's response of the rated POIs can imply what kind of POI(e.g., architecture,academy,history,ecology, *etc.*) he like more or less.

$$
inProperTime(p_i) = \begin{cases} 1 & \text{if current time in the proper period of POI } p_i \\ 0 & \text{otherwise} \end{cases} \tag{3.9}
$$

$$
inProperGender(p_i) = \begin{cases} 1 & \text{if the properGender of POI } p_i \text{ equal the gneder of visitor} \\ 0 & \text{otherwise} \end{cases} \tag{3.10}
$$

$$EqGenre(p_i, p_j) = \begin{cases} 1 & \text{if the genres of } p_i \text{ and } p_j \text{ are equal} \\ 0 & \text{otherwise} \end{cases} \tag{3.11}$$

Finally we use Figure 3.1 shows when the context aware recommender system make suggestion actively.

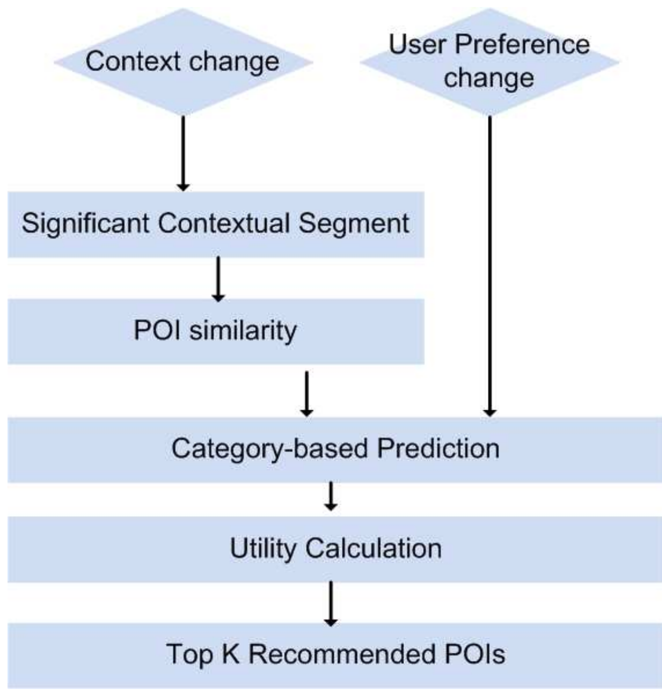

Figure 3.5: Solution Flow Chart

Chapter 4

Implementation and Evaluations

In the chapter, we provide a scenario to demonstrate the context-aware campus spots recommender system, and we use two evaluation metrics to evaluate the usefulness of considering contextual information into recommendation process over ten user.

4.1 Scenario for Existing System

, scenario For mobility, the recommender system with the location-aware devices should be portable with a light weight; we implement the system in a hand-held device, sony UMPC. Then we describe a flow to show how the context-aware recommender system works step by step. At trip beginning stage, a visitor inputs some basic questions such as age, gender, and who in the same trip. The system uses these personal information to build a basic user profile. Figure 4.1 shows the user interface which asks demographic information. The visitor's position is obtained from a GPS receiver,

Figure 4.1: User Input Interface

an external hardware connected to the hand held device, and through the serial port the GPS receiver sends messages to the system every 5 seconds. Synchronously, the user position will be marked on map correctly, the time and weather condition also show in ”weather” field on the screen. According to the user position the nearest POI appears in ”where am I” field on the right side of the screen in yellow circular-shape, meanwhile the user can rate this POI from one star to five stars (the more stars indicates the more favor), and how long he expects to stay. Figure 4.2 shows the nearest POI and weather condition. Moreover, through observing rating distribution with time, we assigned each POI a proper hour to be visited. When a visitor trival along with this system, according to his position the nearby POIs which are proper in current hour will be displaed in green diamond-shape for user to check at his convenice.

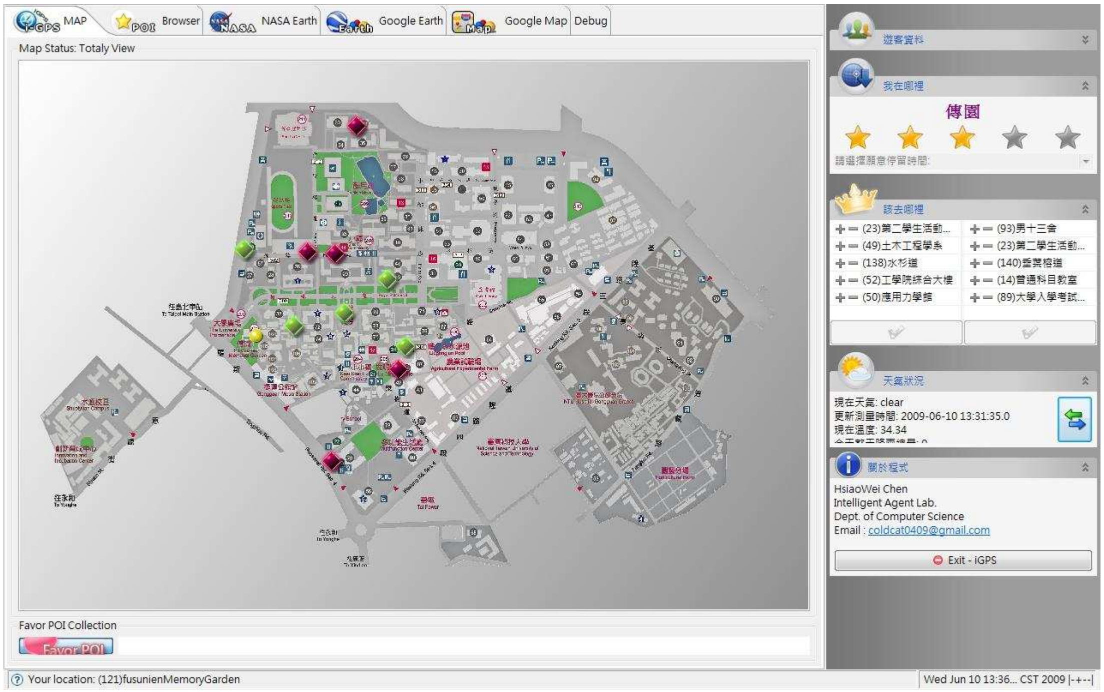

Figure 4.2: The Nearest POI Accoring to the Positon of User

The score action and enviroment change trigger recommendation process begins, Depending on current time and weather, a significant contextual segment will be picked to compute the user-dependent POI similarity. Then the system infers the recommended scenic spots according to user's responses of what had visited in this trip, and attractions information. The top 5 recommended spots will be displayed in green diamond-shape for reference. Figure 4.3 shows recommendations and user response. User can accept(plus sign), reject(minus sign), no comment(by default) the spots in top 5 list. In the following trip, the system will not recommend the spots that the user had rejected. And the accept spots are user's favorite spots which are collected in a list on the bottom of the campus map, and displayed in pink love-shape on the screen. Press up the favor POI button let all favorite spots disappear from the map.

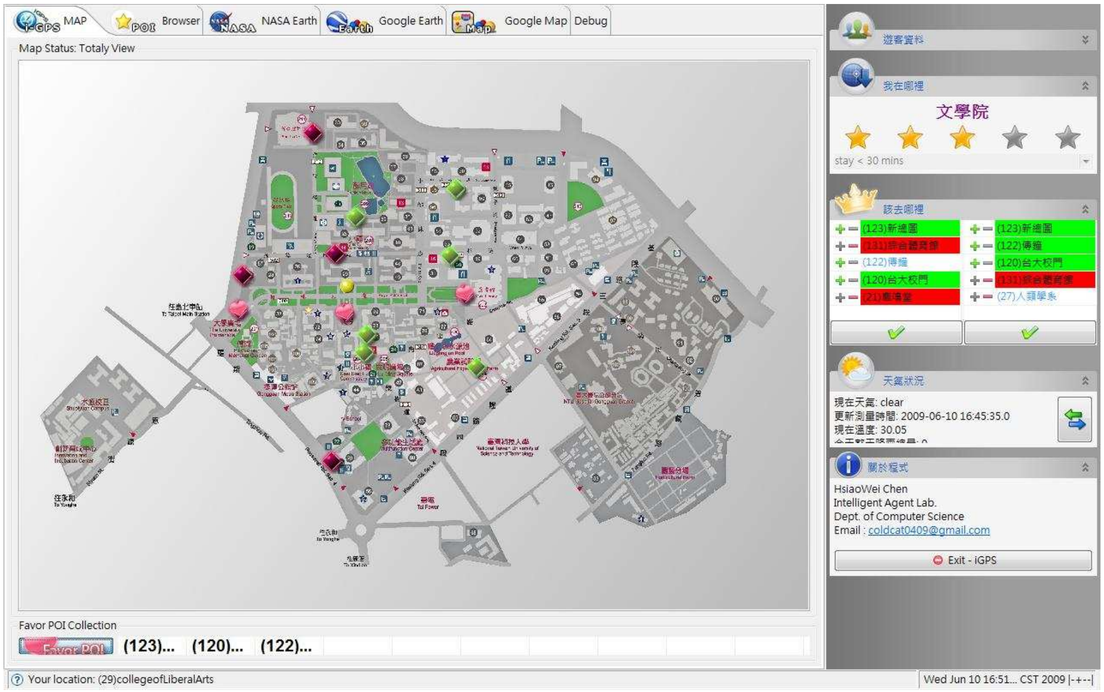

Figure 4.3: The Recommendation Result

While touring, visitors may want an overview of their tour and zoom in a particular part to see the map in detail. There are two guidance modes in this system, the global guiding mode show campus panoramic view, and the local tracing mode zoom in from user coordinates. User can switch them by double click. Figure 4.4 shows the local view.

Since the Computer and Information Networking Center at NTU have already implemented the guidance data about each building (miniGIS [1]), we embed a browser and access the introduction page for each lamdmark. In the overview mode, moving the mouse cursor, the white squares which around in 50 pixels will appear. A white square is a POI, and the visitors can get the more detailed landmark information by click the

[1]source: `http://guide.cc.ntu.edu.tw/ntugis/gis_demo.jsp`

Figure 4.4: The Local View

white square on map directly. Getting guidance contetnt from the Internet can decrease maintenance effort and increase the variety of guidance knowledge. Figure 4.6 shows the panoramic view and indication of introduction page. Figure 4.6 shows the introduction page of Fubell.

4.2 Experiment Evaluation

In this work, we want to know whether the contextual information (time and weather) is useful in making spots recommendation for who is not familiar with NTU campus.

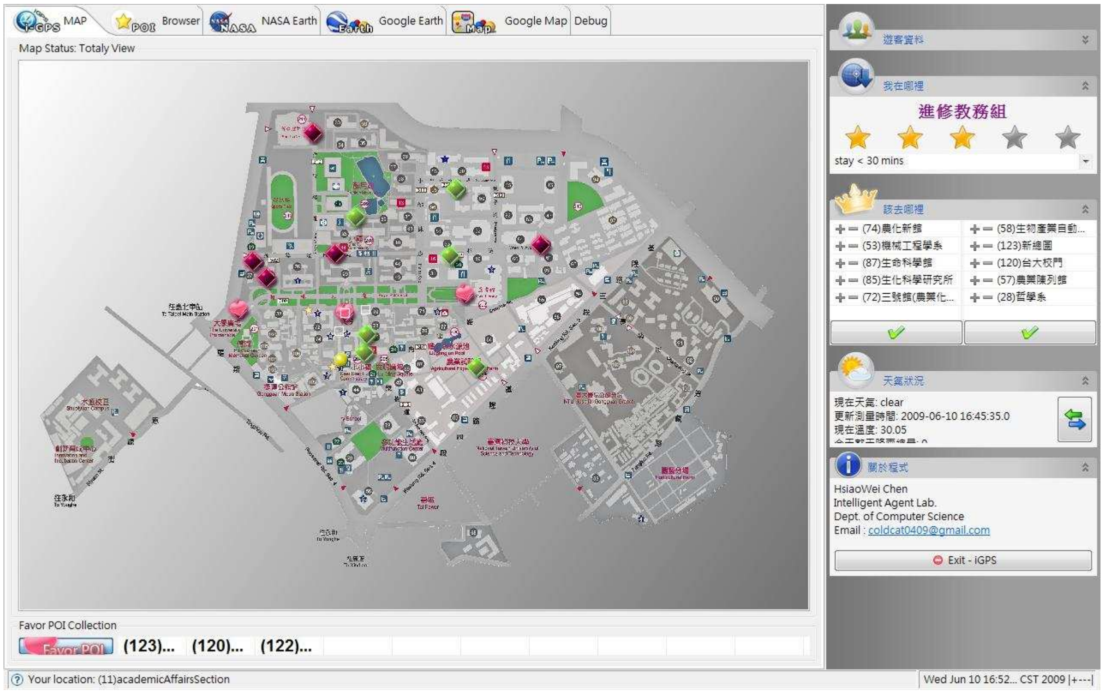

Figure 4.5: The Panoramic View

4.2.1 Experiment Design

In our hypothesis, the target users have no time to plan their trips and have no idea to decide what to visit. We chose 10 people to join this experiment, but they do not know they will visit NTU campus with our system before they take part in. Since the recommendation result will be affected by the user demographic information, we work towards creating a more balanced distribution of age, gender and tourist group.

4.2.2 Evaluation Metrics

Recommender systems research has used several types of measures for evaluating the quality of a recommendation system. In this work we desing two metrics to see whether

Figure 4.6: A Introduction Page about FuBell

the context-aware technique is really improve the recommendation quality in scenic spots domain.

- Acceptance:

  We offer two recommendations lists at the same time, one is produced by taking account of current time and weather condition, but the other does not consider the context issue. Naturally, the premise is that the current context is one of the high-performance contextual segments, or the results of two method will have no difference. After every rating, we ask participants to plus POIs in both lists which attracts him in current situation. Then acceptance percentage of recommendations is computeed over total ratings.

$$Acceptance_{ca} = \frac{|\text{accept}_{ca}|}{|\text{total ratings}|}$$
$$Acceptance_{static} = \frac{|\text{rate}_{static}|}{|\text{total ratings}|}$$

We compares the accept percent of recommendatinos using context-aware collaborative filtering with that using pure collaborative filtering. The bigger accept rate of context-aware recommendations indicates that context-aware technique is useful in making recommendation, since the users do not know which list has context-aware ability.

- Rejection:

  Similarly, we ask participants to minus POIs in both lists which he does not want to see in the following trip. Then rejection percentage of recommendations is computeed over total ratings.

$$Rejection_{ca} = \frac{|\text{reject}_{ca}|}{|\text{total ratings}|}$$
$$Rejection_{static} = \frac{|\text{reject}_{static}|}{|\text{total ratings}|}$$

  We compares the reject percent of recommendatinos using context-aware collaborative filtering with that using pure collaborative filtering. And we think that the smaller reject rate of context-aware recommendations indicates that context-aware technique is useful for spots recommendation.

Finally from Table 4.1 we seven four of ten individuals have higher acception from context-aware recommednations, and nine of ten individuals have lower rejection from

Table 4.1: Evaluation Result

Gender	**Age**	**Group**	**Context**	**Recommender**	**Total Rate**	**Accept**	**Reject**
male	24	alone	10:10-11:20, clear	static	13	14	9
				context-aware		23	4
male	23	alone	14:30-15:50, clear	static	15	22	27
				context-aware		38	12
female	52	family	13:30-14:40, rainy	static	10	15	30
				context-aware		33	12
female	17	friend	15:00-16:10, cloudy	static	17	44	33
				context-aware		40	5
male	27	couple	16:02-16:55, cloudy	static	12	35	23
				context-aware		19	2
female	25	alone	09:05-09:50, rainy	static	16	19	28
				context-aware		28	14
male	29	couple	14:00-15:00, clear	static	12	8	17
				context-aware		7	9
male	33	alone	12:30-13:50, cloudy	static	10	10	9
				context-aware		21	6
female	49	family	09:23-10:30, clear	static	11	5	10
				context-aware		10	3
female	18	friend	15:20-16:20, clear	static	15	11	3
				context-aware		17	5

context-aware result. The evaluation indicates that context-aware technique is useful in making recommendation for campus scenic spots.

Chapter 5

Conclusion

The contribution of this thesie is that we implement a system which offers the context-aware campus spots recommendations for who has no time to plan his travel schedule. The system dynamically adjusts recommendations on the basis of spot attributions and visitors preference under different context. The important aspects of context include in this work are: where the visitor is, what time it is, whether it rains, how visitor feel about current trip, and who the visitors is with. We propose an innovated soluction to realize a context-aware campus spots recommender system, and integrate several technologies, including recommender algorithm, context-aware ability, user modeling, and ontology. In particular, we assume that the user is a new one who has no preference on any POIs and no user of the community, therefore user-based collaborative filitering does not work well in this situation. We use stereotype to build a rough user model quickly and observe the interaction between user and system to implicitly predict user's preference. Moreover, when vote predicting we take individual responses and the opinions of who has similar demographical information into account, this can alleviate the rating sparse problem. In other words, the user's preference is empty in the begin, and it refines after tracking the user behavior and feedback. The benefit of this approach is that the user's profile is dynamic and constantly update without disturbing the visitor. The Implicitly rating plays a critical role in context-aware recommender system, because it is too verbose if we ask users grade the same POI every ten minutes.

We choose ontology rather than database to construct POI model and time model, since an ontology-based knowledge base provides the abilities of knowledge conceptualization and modification flexibility. The common database can tackle the POI properties, but it is difficult to conceptualize the relationships between POIs. The ontology digitalize guidance experience and conceptualize the relationships among objects, in this work, we have built a location model about all spots in NTU campus, includes the positions, category and introduction documents. Furthermore an ontology-based system provides a better ability of information inference; Given a visitor's position, the system show neighboring spots which is popular in a specific hour of day.

In this work, the visitors could experience recommendations depending on their personal data and the environment conditions. With up-to-date user responses in the trip, the system provide a personalized suggestion, moreover, the recommendation result varies with different time and weather condtion. The high performance contextual segments deciding alogrithm helps systme to deicde which combination of time and weather is better with respect to rating prediciton on the campus travl domain. This is a interesting area of applications where context-awareness system can provide recommendations to new uses. We report on cooperative efforts between context-aware technique and item-based collaborative filtering. Through the experiment evaluation, we design three evaluation metric to verify the performance of the system. Finally, the experiment results show that integrated method outperforms an existing social-filtering method in the domain of campus spot recommendations on a dataset of 83 spots ratings collected over 6 users.

Bibliography

[1] G. Adomavicius and T. Alexander. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17(6):734–749, 2005. Member-Gediminas Adomavicius and Member-Alexander Tuzhilin.

[2] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145, 2005.

[3] G. Adomavicius and A. Tuzhilin. Multidimensional recommender systems: A data warehousing approach. In *WELCOM '01: Proceedings of the Second International Workshop on Electronic Commerce*, pages 180–192, London, UK, 2001. Springer-Verlag.

[4] G. Antoniou and F. van Harmelen. *A Semantic Web Primer*. MIT Press, April 2004.

[5] L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Intrigue: Personalized recommendation of tourist attractions for desktop and handset devices. In *Applied Artificial Intelligence*, pages 687–714. Taylor and Francis, 2003.

[6] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with gps.

[7] C. Basu, H. Hirsh, and W. Cohen. Recommendation as classification: Using social and

content-based information in recommendation. In *In Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 714–720. AAAI Press, 1998.

[8] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. pages 43–52. Morgan Kaufmann, 1998.

[9] A. Chen. Context-aware collaborative filtering system: Predicting the user's preference in the ubiquitous computing environment. In *LoCA*, pages 244–253, 2005.

[10] H. Chen, T. Finin, and A. Joshi. An intelligent broker for context-aware systems. In *UbiComp 2003: Adjunct Proceedings of the Fifth International Conference on Ubiquitous Computing*, pages 183 – 184, Seattle, Washington, USA, October 2003. UbiComp.

[11] H. Chen, T. Finin, and A. Joshi. A context broker for building smart meeting rooms. In C. Schlenoff and M. Uschold, editors, *Proceedings of the Knowledge Representation and Ontology for Autonomous Systems Symposium, 2004 AAAI Spring Symposium*, pages 53–60, Stanford, California, March 2004. AAAI, AAAI Press, Menlo Park, CA.

[12] H. Chen, T. Finin, and A. Joshi. *Ontologies for Agents: Theory and Experiences*, chapter The SOUPA Ontology for Pervasive Computing, pages 233–258. Whitestein Series in Software Agent Technologies. Springer, July 2005.

[13] H. Chen, F. Perich, T. Finin, and A. Joshi. SOUPA: Standard ontology for ubiquitous and pervasive applications. In *First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04)*, pages 258–267, August 2004.

[14] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Developing a context-aware electronic tourist guide: Some issues and experiences. In *Proceedings of the SIGCHI conference on Human factors in computing systems(CHI 2000)*, pages 17–24, New York, NY, USA, 2000. ACM.

[15] K. Cheverst, K. Mitchell, and N. Davies. The role of adaptive hypermedia in a context-aware tourist guide. *Communications of the ACM*, 45(5):47–51, 2002.

[16] L. Console, S. Gioria, I. Lombardi, V. Surano, and I. Torre. Adaptation and personalization on board cars: A framework and its application to tourist services. In *AH*, pages 112–121, 2002.

[17] A. K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.

[18] J. Froehlich, M. Y. Chen, I. E. Smith, and F. Potter. *Voting with Your Feet: An Investigative Study of the Relationship Between Place Visit Behavior and Preference*. 2006.

[19] G. Fu, C. B. Jones, and A. I. Abdelmoty. Building a geographical ontology for intelligent spatial search on the web. In *Databases and Applications*, pages 167–172, 2005.

[20] T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Deventer, The Netherlands, 1993. Kluwer Academic Publishers.

[21] G. Karypis. Evaluation of item-based top-n recommendation algorithms. pages 247–254, 2000.

[22] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. Grouplens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.

[23] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[24] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Online, 2001.

[25] R. Oppermann and M. Specht. A context-sensitive nomadic exhibition guide. In *HUC: handheld and ubiquitous computing*, pages 127–142, 2000.

[26] M. J. Pazzani. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408, 1999.

[27] C. Pils, I. Roussaki, and M. Strimpakou. *Location-Based Context Retrieval and Filtering*. 2006.

[28] S. Poslad, H. Laamanen, M. Rainer, N. Achim, B. Phil, and Z. Alexander. Crumpet: Creation of user- friendly mobile services personalised for tourism, 2001.

[29] E. Rich. User modeling via stereotypes, January 1979.

[30] N. M. Sadeh, F. L. Gandon, and O. B. Kwon. Ambient intelligence: The MyCampus experience. Technical Report CMU-ISRI-05-123, School of Computer Science, Carnegie Mellon University, July 2005.

[31] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA, 2001. ACM.

[32] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. CF-sparisity:158–167, 2000. POI model : Association Rule.

[33] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. T. Riedl. Application of dimensionality reduction in recommender systems a case study. In *In ACM WebKDD Workshop*, 2000.

[34] B. M. Sarwar, J. A. Konstan, A. Borchers, J. Herlocker, B. Miller, and J. Riedl. Using filtering agents to improve prediction quality in the grouplens research collaborative filtering system. pages 345–354, 1998.

[35] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 85 – 90, Santa Cruz, CA, US, 1994.

[36] A. Schmidt, K. Asante Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de Velde. Advanced interaction in context. In *HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 89–101, London, UK, 1999. Springer-Verlag.

[37] A. Schmidt, M. Beigl, and H.-W. Gellersen. There is more to context than location. *Computers and Graphics*, 23(6):893–901, 1999.

[38] I. Schwab and W. Pohl. Learning information interest from positive examples, 1999. modifed mechine learning.

[39] M. v. Setten. Experiments with a recommendation technique that learns category interests. In *ICWI*, pages 722–725, 2002.

[40] M. v. Setten, S. Pokraev, and J. Koolwaaij. Context-aware recommendations in the mobile tourist application compass. volume 3137, pages 235–244. Springer Berlin / Heidelberg, 2004.

[41] Y. Takeuchi and M. Sugimoto. A user-adaptive city guide system with an unobtrusive navigation interface. *Personal and Ubiquitous Computing*, October 2007.

[42] R. Want, A. Hopper, V. F. ao, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, January 1992.

[43] J. Ye, L. Coyle, S. Dobson, and P. Nixon. A unified semantics space model. In J. Hightower, B. Schiele, and T. Strang, editors, *LoCA*, Lecture Notes in Computer Science, pages 103–120. Springer, 2007.

[44] U. L. Yuhana, L.-l. Chen, J. Y.-j. Hsu, and W.-r. Jih. An ontology based approach for searching neighborhood building. In *Proceedings of The Third International Seminar Information and Communication Technology Seminar (ICTS07)*, pages 106 – 112, Surabaya, Indonesia, September 11 2007.