LePlaza: An Event-Centered Social Networking Platform with Location Based
Personalized Recommendation Service

by

Cong Lin

B.Sc., Tsinghua University, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2011

© Cong Lin, 2011
Abstract

Web based social networking services have afforded opportunities for people to enrich their social lives by building new relationships and sharing various kinds of information. While many people are interested in questions such as “what interesting social events are happening nearby that I can join?”, few existing social networking sites have successfully provided easy solutions for people to explore and organize real social activities.

This thesis represents LePlaza platform, an effort to offer an event-centered social networking platform through which the users can share real social events as well as scheduling their own events. To serve targeted events that are more likely to be of interest to a specific individual, a content-based user-interest model is proposed according to the assessment of the user’s historical activity records and the description of the events. An intelligent recommender system built upon the user-interest model is developed to make personalized recommendations on restaurants as a particular case study in which events are parameterized by user’s location, and the results are visualized in an interactive manner. Experiments and user studies show that the proposed model and algorithm effectively reflect the user’s personal preferences with satisfying recommendation accuracy for the case of dining events.
# Table of Contents

Abstract........................................................................................................................................... ii  
Table of Contents .......................................................................................................................... iii  
List of Tables ...................................................................................................................................... v  
List of Figures ..................................................................................................................................... vi  
Acknowledgements .......................................................................................................................... vii  
Dedication .......................................................................................................................................... 1  

## Chapter 1: Introduction .................................................................................................................. 1  

1.1 Background and Motivation ....................................................................................................... 1  
1.2 Thesis Goals and Contributions ............................................................................................... 4  
1.3 Structure of the Thesis .............................................................................................................. 5  

## Chapter 2: Related Work ............................................................................................................... 7  

2.1 User Profiling and Interests Modeling ...................................................................................... 7  
2.2 Recommender System .............................................................................................................. 10  
2.3 Existing Personalized LBS Systems ......................................................................................... 13  

## Chapter 3: LePlaza Platform – System Architecture .................................................................. 15  

3.1 Overall Architecture ................................................................................................................ 15  
3.2 Friend Management ................................................................................................................ 18  
3.3 Event Management ................................................................................................................ 19  

## Chapter 4: LePlaza Platform – The Recommendation System .................................................. 20  

4.1 Entity Representation .............................................................................................................. 20  
4.2 User Interest Model ................................................................................................................ 26
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3 Personalized Recommendation</td>
<td>30</td>
</tr>
<tr>
<td>4.4 Summary</td>
<td>33</td>
</tr>
<tr>
<td>Chapter 5: LePlaza Platform – Visualization</td>
<td>34</td>
</tr>
<tr>
<td>5.1 Dining History</td>
<td>34</td>
</tr>
<tr>
<td>5.2 Preference Analysis</td>
<td>35</td>
</tr>
<tr>
<td>5.3 Recommendation Results</td>
<td>37</td>
</tr>
<tr>
<td>Chapter 6: Evaluation</td>
<td>39</td>
</tr>
<tr>
<td>6.1 Effectiveness of Keyword Vector</td>
<td>39</td>
</tr>
<tr>
<td>6.2 Determine Lubricating Factor</td>
<td>43</td>
</tr>
<tr>
<td>6.3 User Study</td>
<td>46</td>
</tr>
<tr>
<td>Chapter 7: Conclusion</td>
<td>50</td>
</tr>
<tr>
<td>7.1 Contributions</td>
<td>50</td>
</tr>
<tr>
<td>7.2 Future Work</td>
<td>51</td>
</tr>
<tr>
<td>Bibliography</td>
<td>53</td>
</tr>
<tr>
<td>Appendix: Original Data of Restaurant Recommendation User Feedback</td>
<td>56</td>
</tr>
</tbody>
</table>
List of Tables

Table 4.1: Examples of Dining Activity ................................................................. 21
Table 4.2: Representation of Restaurant Entity ...................................................... 25
Table 4.3: Representation of Restaurant Review .................................................... 25
Table 4.4: Recommendation Algorithm .................................................................... 32
Table 6.1: Intra-category Similarities ...................................................................... 41
Table 6.2: Inter-category Similarities ...................................................................... 42
Table 6.3: Precision Comparison ........................................................................... 48
Table 6.4: Recall Comparison ................................................................................ 48
List of Figures

Figure 1.1: Evolution of Location Based Search ................................................................. 4
Figure 2.1: Example of Topic Hierarchical Tree ................................................................. 9
Figure 3.1: LePlaza User Interface .................................................................................... 16
Figure 3.2: LePlaza System Architecture ............................................................................ 17
Figure 3.3: Friend Tracking & Route Planning ................................................................. 18
Figure 4.1: An Example of Weighted Keyword Vector ....................................................... 25
Figure 4.2: Representation of Restaurant Entity ............................................................... 26
Figure 4.3: An Example of Dining History Tree ............................................................... 27
Figure 4.4: An Example of User Interest Tree .................................................................. 28
Figure 5.1: Dining History Visualization .......................................................................... 34
Figure 5.2: Summary Page of Dining History ................................................................... 36
Figure 5.3: Detail Page of Dining History .......................................................................... 36
Figure 5.4: Customized Recommendation Result ............................................................. 37
Figure 5.5: Location Based Recommendation Result ....................................................... 38
Figure 6.1: Measurement D Varies with \( \alpha \) .................................................................. 46
Figure 6.2: Precision Comparison ..................................................................................... 48
Figure 6.3: Recall Comparison .......................................................................................... 49
Acknowledgements

- First of all, I would like to offer my enduring gratitude to my supervisor Dr. Son Vuong for his guidance. In the journey of two years, I have got great opportunities from him and his lab, and his support has always been a great motivation for me.

- I would also like to give thanks to the second reader of my thesis, Dr. Eric Wohlstadter, and my lab mates: Andrew, Victor, Shahed, and all others for their kind help.

- Special thanks are owed to my parents for their endless support in my life.
Chapter 1: Introduction

Social networking is a hot topic in internet computing area today. Rich contents can be posted and shared by people on social networking websites, including status, articles, photos, music, and videos. However, these are not typically what people do in their real social lives, where people are usually connected by various social events such as parties or sports games. Therefore, the first question is, can those social events be managed and shared by people on social network websites as conveniently as other stuffs?

In addition, when talking about social events, people are often interested in questions like “Where does the event X take place?” or “What is happening at location Y?” The location based services are supposed to deal with such questions. However, the task of selecting some really interesting items among the huge number of choices returned by a location based search is left to users, because the traditional location based services have no idea of user’s personal preferences. So the second question to be answered is, how to find upcoming events that are interesting to a specific user? In this thesis, a location-based, event-centered social networking platform called LePlaza is presented, aiming to give answers to the questions above.

1.1 Background and Motivation

The introduction of the internet and web-based social networking sites such as Facebook and Twitter, has provided good answers to questions with “who”. People build relationships with friends in their real lives and strangers from every corner of the internet in
the cyberspace, where they chat, share status and photos, and play online social games together.

Despite that many social networking sites have brought enormous fun to people’s online social lives, few have been successfully provided a media that ties people’s online social lives to their real lives, a platform that allows people to share, schedule, organize and get notified of real-life social events like an outdoor hiking or a live concert, with ease.

To find the answers to questions with “where”, people could acquire a city tour guidebook or they would instead scan through the internet and search pages for keywords they are interested in. However, one may despondently find that either some of the places are not included in an old dated print guide or an upcoming event returned by traditional search engine is actually far away from where he lives. Traditional web search works as shown in Figure 1.1(a).

The coming out of Location Based Service (LBS) has made such geographically-related questions easier. Location Based Service is the use of certain mobile technical approaches through the internet/mobile network to access user’s location information (latitude and longitude coordinate), and provides users with corresponding value-added services through the electronic map platform [1] [2]. Examples are restaurant finders, buddy trackers, applications in the areas of mobile marketing and mobile gaming. The attractiveness of LBSs comes from the fact that users are not required to enter location information manually but are automatically pin-pointed and tracked thanks to nowadays mobile devices. Some local event search services such as Eventful [3] and Zevent [4] help to find out public facilities and business in the surroundings and what is going to take place nearby, as shown in Figure 1.1(b). The result is usually presented with a list of upcoming events and their
addresses. However, as a classical music concert may not be attractive to a fan of rock and roll, the conventional location based search engine hardly knows the differences among people with different tastes, because there is insufficient way for them to express their personal preferences during search. Providing the same result to people with different tastes never satisfies all.

Known as Context-Aware Service, an advanced location based service integrates personalized information and more targeted context services as shown in Figure 1.1(c). One very interesting class of context aware services are so called personalized applications [5], which suggests that “how well” an item fits a specific user. In the social life domain in this thesis, personalization means the use of knowledge of the user’s personal preference on various social events as well as the location of the user to find place/events of interest with minimal effort on the user’s part.

Most of the personalized applications make use of a key technology called recommender system [6]. The recommender system was originally developed to address data overload issue, which is extremely important to applications running on hand held devices as the small screen and limited interactions make it difficult for navigating and searching trough large space of data. It has greatly contributed to success of e-commerce companies as Amazon and EBay, functioning well to recommend relevant products to consumers.

However, despite the huge success the recommender system has made in commercial goods recommendations, few system delivering location based, personalized social activity information exists today.
1.2 Thesis Goals and Contributions

The overall goal of this thesis is to design, develop, implement and evaluate a location based social networking platform prototype, on which users can publish, schedule, share, search and join various of social events. The aim also includes that the operations and resultant social events should be tailored to the user’s personal preference and tries to make him feel in control of his experience in a visual manner.

This goal will then be realized on the specific case of events - dining activities in Vancouver, resulting in a service that would help a user view his dining activities and make dining decisions according to his own historical dining records. The reason of choosing dining event as a particular case in the thesis is we believe that dining event is one of the most popular social events in people’s social lives, and moreover, it is very convenient to retrieve dining-related data such as restaurants information and customers’ reviews from criticism websites.
The main contributions of this thesis are:

First, a location-based and social-event centered platform is designed and implemented. Some of the technical goals of the platform are:

- User can publish, view, attend and invite other users to join social events;
- User can perform an event search on his or her location and optional locations;
- User can be presented the search result as selectable markers on a map and items in a list;
- User can perform group operations on events and users.

Second, a new model of representation of user’s personal preferences on social events based on user’s historical social activities is presented. There are variety types of social events in the system, including education, sports, music, and so on. In this thesis a preference model of dining events is built and implemented, but the model itself is expected to be easily applied to other types of events in system. A recommending algorithm is run on pre-built preference model, aiming to give recommendations on local restaurants which meet the user’s taste.

Third, the user’s history of social activities, preference model as well as recommendation results are visualized in an interactive and playful manner by using e-maps, graphs and diagrams.

1.3 Structure of the Thesis

Chapter 2 discusses related work with respect to user interest modeling and personalized recommendation, as well as the state of art of location based social platforms.
Chapter 3 gives an overall introduction to our event-centered social network platform, LePlaza, including system architecture, function modules and other technical details. Chapter 4 describes the steps of design and implementation of user preference model for dining events. A personalized recommendation algorithm running on the model will then be proposed. In Chapter 5, a user-interface is presented to visualize the model and results generated in previous chapter. Evaluations are made in Chapter 6. System parameters are adjusted based on experiments and final user experience evaluations are studied. The last chapter concludes the thesis.
Chapter 2: Related Work

The most important issue for many personalized applications is to construct a computational model for each individual user to predict his preferences for upcoming events [7]. An accurate user profile is vital to the system’s success as it largely depends on the ability of the model to represent the user’s interests. In this chapter approaches of user profiling are firstly discussed, followed by recommendation algorithms running on the profile. Finally some state of the art relevant applications are presented.

2.1 User Profiling and Interests Modeling

A user profile is a description of the user containing the most important facts about him or her [8]. User profiling is a process of deriving unobservable features about users from observable information given by them or their actions in system. The most common contents of user profile include: user preferences and interests; the user’s individual characteristics; user behavior patterns; the user’s knowledge and skills; the user’s goals.

A user preference and interests model is a subset of a user profile as it is only concerned with modeling the following two aspects with respect to a user [7]: First, it models the interests of a user in some qualitative objects or concepts and second, it measures his preferences in these interests by some quantitative values. The two most common representations of user interests are keyword-based model and topic hierarchical model.

2.1.1 Keyword-based Interests Model

In keyword-based models, user interests are represented by weighted vectors of keywords. Weights traditionally represent the relevance or importance of the keyword for the
user or within the topic. The vector of keywords can be explicitly provided by the user, or they can be extracted from the description of products that user used to buy or view, the articles or Web documents visited by the user during browsing, etc.

Amalthaea [9] is a typical system that creates keyword profiles by extracting them from web pages. The keywords are weighted with the widely used TF-IDF weighting scheme [10] and compared to the profile using the cosine formula [10]. Only the documents with vectors that are closest to the user profile are passed on to the user. Letizia [11] and Fab [12] did similar work in different application scenarios.

In the personalized book recommendation system described in [7], each book is represented by a list of keywords referring to the categories and scope of the book. User’s interest with respect to certain book category is then proportional to the frequency that user browses books containing the category keyword. Similar work can be found in PEA [13], which builds keyword-based profiles using terms extracted from the user’s webpage bookmarks. The difference is that for each bookmark there is a set of keyword vectors generated. The rationale behind this is that the user’s interests in multiple areas can be the combinations of sets of vectors under corresponding bookmarks.

The keyword-based model is suitable for building user profile from unstructured data such as the description of an item or text in a document. The limitation is its inherent polysemy problem: the keyword “apple” could have totally different meanings to a nutritionist and an electronic device fancier.
2.1.2 Topic Hierarchical Interests Model

Topic hierarchical model is usually represented by conceptual nodes and relationship between these nodes. Each node in the hierarchy represents a topic of interest for a user, which is defined by a set of representative weighted words. A typical topic hierarchy can be seen in Figure 2.1. Some topics can have dozens or hundreds children while others may have few or none. In Sensus search project [14], the hierarchy contains approximately 70,000 nodes in total. Because building a deep and broad hierarchical model is mostly an expensive and manual work, profiles are often based on subsets of existing classifications.

![Figure 2.1: Example of Topic Hierarchical Tree](image)

A lot of research has been carried out using topic hierarchies. Although no tree is created, the personalized search engine proposed in [15] introduced the concept of hierarchy by using hierarchical matrices instead to represent user’s hierarchical relationship in his search history. In [16], [17], [18] interest trees are built up and each tree node is represented by a topic-weight pair. The user interests are then captured by a set of preference-vectors, which store the path information and node relationship from root node to a leaf node. These
approaches successfully address polysemy issue, however, they require every object has an explicit topic. In addition, the simple form of “topic-weight” is not powerful enough to represent complicated user interests.

As an attempt to combine keyword-based model and hierarchical model, in the e-learning system presented in [19], each leaf learning topic node is attached a set of weighted keywords, which are extracted from user’s learning materials. A user interest score is then computed by the sum of the score from hierarchy part and the one from keywords part. It is the closest work to our interest model in the thesis. The difference is that in our model, a list of keywords is attached to every topic node not only to the leaf nodes.

In summary, the good part of a hierarchical model is that it is able to store semantic relationship, which gives us many advantages over a flat structure, as we can utilize the inherent relationships between the nodes of the hierarchy to better understand the user’s interest on a given context. The bad is that it is insufficient when dealing with unstructured data, where no explicit hierarchical structures are available. Therefore, it is believed that a hybrid model, which combines hierarchies and keyword vectors, would be more powerful when constructing interest model from mixed data sources.

2.2 Recommender System

Recommender systems have been an active research area for more than a decade, and many recommendation approaches with distinct strength have been proposed. In the literature, two types of recommendation approaches have been suggested: content-based filtering [20] and collaborative filtering [21].
2.2.1 Content-based Filtering

Content-based filtering systems analyze the content of items unknown to the user, compare them with pre-built user profile which is a representation of user’s interests, and then identify items that are of particular interest to the user [22]. For example, a content-based music recommendation system will typically rely on information such as style, instrument, player, and composer and will match these against the learned user preferences model in order to select a set of promising albums. A real case is Amazon.com [23]. Part of the Amazon’s product recommendation results comes from its content-based feature, “favorites”. These favorites are either calculated by keeping track of the categories of items that users purchased or may be specified by the user. More work can be found in [24].

Content-based filtering is very efficient in text-intensive domains, and it has the advantage of immediate recommendations, i.e., no feedback from other users about the items is necessary. However, some well-know shortcomings of this approach are discussed in [25], [26], and [27]. Firstly, users can only be recommended with items that are similar to the ones they visited in the past. Secondly, sometimes it is difficult to formalize user perception and preference. Thirdly, the system will need to work with a sufficient amount of data.

2.2.2 Collaborative Filtering

The rationale of Collaborative Filtering (CF) is the fact one is likely to enjoy an item if his friends who share the similar tastes recommend it. Rather than determining the similarity between item and user preferences (as in Content-based approaches), CF analyzes users’ historical ratings to identify similarities between users, and then generates new recommendations based on like-minded users’ preferences.
A prototype of an academic event recommender system is discussed in [28]. The system constructs a researcher-academic event participating matrix, based on which similarity between any two researchers is computed. To make recommendations to a target researcher, a group of the most similar researchers is first selected and the rank of upcoming events is then determined by their aggregating ratings. Similar work is done in [29], [30]. A flaw of this method is that it requires users’ feedback from events that haven’t taken place yet.

To address the problem that how to use the partial rating history to predict ratings for the remaining items that the users have yet to experience, [31] uses pair of event in past \((E_i, E_j)\) to represent user preferences, where \(E_i\) is a liked event and \(E_j\) is a disliked event, and attempts to find a ranking function to the unknown event list that best captures past feedback (e.g. \(E_i\) is preferred over \(E_j\)) from the user. The method, however, is not really collaborative as it doesn’t take the impact from other similar users into consideration.

The strength of collaborative filtering is that it is completely independent of any machine readable representation of the objects being recommended, and works well for complex objects such as movies in which people’s tastes and preferences vary [32]. A disadvantage of CF is the system is depended on many active users to ensure a sufficient amount of ratings and a proper distribution of the ratings across all the items.

In summary, both content-based and collaborative approaches have their strengths and weakness. In our particular case where dining events are to be recommended, a content-based method is preferred as the features of the contents are relatively easy to be extracted from a given data source and it is believed one’s preference in food is mostly determined by his personal taste rather than other’s opinions.
2.3 Existing Personalized LBS Systems

Four personalized location based services are investigated in this section, including two web-based services, one social networking application and one mobile phone application.

2.3.1 Loopt

*Loopt* [33] is a mobile phone application, as shown in Figure 2.2, which allows users to see the nearby buzz about the places and on-going events around them and where their friends are.

The application is inspired by *Google Latitude*, which is a simple mobile location-based application that allows the user to see the positions of his friends on Google map. *Loopt* adds more social elements to it by marking interesting places with highlighted features on map and encourage users to share comments. The recommendations on places are based on popularity of the place. Thus there is no personal preference considered.

2.3.2 GeoLife Project

As a part of the *GeoLife* Project developed at Microsoft, [34] presents a web-based tool that makes two types of recommendation: (1) Location recommendation given some activity query and (2) Activity recommendation given some location query.

By mining GPS data and available user comments, the system is able to build up a location-activity correlation matrix. Recommendations can then be made due to the fact that certain activities are often performed at certain locations. No personal data is used in the system.
2.3.3 LivesGEO

LivesGEO [48] is a location based multimedia sharing and tagging mobile application developed by Network and Internet Computing Lab, Computer Department of UBC. Pictures and videos can be uploaded via mobile phone and marked on the map where people took them. Users can also search for contents shared by others on the map. LivesGEO and LePlaza project in this thesis complement each other: People schedule events on LePlaza, and share exiting moments of the events they attended using LivesGEO.

2.3.4 Eventer

The Eventer [26] is a Facebook application that recommends upcoming concert events to a specific user. The application uses an IP-GeoLocation mapping to get location data and retrieves concert events from a third music website. The combination of both content-based and collaborative methods is used to compute similarities between user and event, as well as user and user. The limitation is a user has no control of his location which is only determined by his IP address, and content-based similarity calculation relies on the data format from the specific music website.
Chapter 3: LePlaza Platform – System Architecture

In this thesis, we developed LePlaza, a location based online social network which provides event-centered services such as social events publishing and searching, personalized event recommending, etc.

The core concept of LePlaza is event. People are connected by events they attended and are going to attend. An event could be any kind of social activity like an academic conference, a weekend party or a soccer game on campus. Using the location information collected from mobile clients, LePlaza allows user to get real-time info of social events happening nearby, to know which friends are going to attend the event and to search potential participants nearby who have the similar interests.

This chapter gives an overview of our LePlaza platform, including the system architecture and the introduction to each function module. More details about personalization module, which is the main work of this thesis, will be discussed in later chapters.

3.1 Overall Architecture

A user interface of LePlaza after the user logged in is shown in Figure 3.1. Like many mainstream social networking sites, the front page contains a navigation of functional links, a list of friends, and friends’ news feed from left to right, respectively.
A high level view of the system consists of four functional components: User identification management, friend management, event management and messaging module. The complete system architecture is shown in Figure 3.2. ID management component keeps the user’s profile information and validates when user login. Online chat and offline message are both supported by MSG management component, as well as the update of friends’ news feed. The content in dash box is the core work of this thesis, which will be described in detail in the following chapters.
Figure 3.2: LePlaza System Architecture
3.2 Friend Management

Friend management module provides a set of basic operations such as view, invite, search, etc. User can use a keyword-based search engine to search other user’s profiles which are open to public, view their information and send new friend invitations. It is desirable that all the basic functions can be performed on a basis of group. This is useful when a user wants to schedule an event shared by a group of people. The group module allows the user to create and join groups, invite others and send messages to whole group or any selected members.

Another feature of this module is it allows user to keep a track of his friends on the map. It also has a route planner to provide user a route to any selected friends. Figure 3.3 shows the interface for friend tracking.

![Figure 3.3: Friend Tracking & Route Planning](image_url)
3.3 Event Management

Like friend management module, event management component also provides some basic event operations in its Public Event sub-module. The event search is location-based and a search range can be specified by the user. Only events that match search condition and are within the given range are returned. In addition, more filters can be applied to search result such as event category and date.

Any user can create new events and send event invitations to any selected friends either right after the events are created or later. A friend who decided to attend the event will be marked on an event-view map where all event-related people can be seen.
Chapter 4: LePlaza Platform – The Recommendation System

As a specific case of social event, user’s dining activities are studied in this chapter. Because the details of recommendation systems differ based on the representation of items, this chapter first discusses item representations. Next, a user interest tree is built upon the data representations. Finally a personalized restaurant recommendation service using user interest tree is suggested.

4.1 Entity Representation

In order for the system to analyze item description to identify items that are of particular interest to the user, presentations of two types of entities in system must be studied: the presentation of user’s historical dining activity, which reflects his dining preferences, and the presentation of restaurant entity, which contains the feature and description of the restaurant.

4.1.1 Historical Dining Activity

After a user having his meal, he could add a record of historical dining activity to system database through the dining activity uploading interface of LePlaza. A table called Dining_Activity_Table is used by the system to store dining activities for all users. Table 4.1 shows the table with a few records that describe three dining activities.
Table 4.1: Examples of Dining Activity

<table>
<thead>
<tr>
<th>ID</th>
<th>uid</th>
<th>rid</th>
<th>date</th>
<th>cost</th>
<th>rating</th>
<th>with_friend</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>14</td>
<td>bobs-subss-richmond</td>
<td>2011-05-04</td>
<td>12</td>
<td>4</td>
<td>4;5;</td>
</tr>
<tr>
<td>1002</td>
<td>14</td>
<td>toshi-sushi-vancouver</td>
<td>2011-07-11</td>
<td>15</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>1003</td>
<td>15</td>
<td>red-robin-burnaby</td>
<td>2011-03-20</td>
<td>19</td>
<td>3</td>
<td>10;</td>
</tr>
</tbody>
</table>

Each record has an ID as unique identifier. The rest of fields of a record are: user id (uid), restaurant id (rid), dining date, dining cost, user’s rating and friends who had meal together.Uid and rid are foreign keys for User_Table and Restaurant_Table in database respectively. Rating is an integer from 1 to 5 with minimum of 1, specified by user, indicating the satisfaction for the meal. The with_friend field contains a NULLABLE list of friend ids.

4.1.2 Restaurant Data Collection

There are two types of restaurant data, structured and unstructured. The structured data includes most information of the restaurant such as name, category, address, hours, and average rating, etc., whereas the unstructured data is users’ reviews.

All restaurant data is from Yelp [35], which is a social networking, user review and local search website [36]. Yelp offers online local search includes what the user is seeking (e.g. a restaurant) and the location from which the search is performed. The returned result usually contains both structured data and unstructured data (reviews).
Yelp also provides a set of APIs to complete the same task as its online local search. However, there is a limitation of using Yelp API to collect restaurant data, which is, the Yelp API only returns the most recent three reviews of a restaurant. This is not good enough for us to extract features for the restaurant from the reviews. Therefore, a web crawler is written to gather all users’ reviews for a target restaurant. Steps to collect all restaurant data in Vancouver:

Step I Call Yelp local search API with search parameters ‘city = Vancouver’ and ‘type = restaurant’. A list of restaurants is returned with all desired structured data and a page url for each restaurant on Yelp website. Store the structured data in Restaurant_Table.

Step II For each restaurant page url, run the crawler to crawl all the reviews associated to the restaurant, and store them in Restaurant_Review_Table.

We have collected 816 restaurants in Vancouver and 12,800 reviews in total.

4.1.3 Restaurant Entity

The representation of restaurant entity consists of the representation of structured data and unstructured data. The structured data is stored in Restaurant_Table and some of the important fields of the table is shown in Table 4.2, where category is the style of the restaurant, and url is the link to the detail page of the restaurant where reviews can be collected. Other fields that are not shown in the table include address, photo, phone, price, highlight_words of restaurant, etc.
The unstructured data, which is user’s review, is firstly store in _Resturant_Review_Tables_ shown in Table 4.3, where _ID_ is a unique identifier for the record, _rid_ is the restaurant id, _rev_rating_ is the user’s rating for the restaurant. For each restaurant, a weighted keyword vector is attached to describe the features of the restaurant. The keyword vector is generated from the stored reviews.

In order to compute a weighted keyword vector for each restaurant, a keyword dictionary needs to be generated first. The keyword dictionary, which is extracted from the contexts of reviews, contains all the keywords that will appear in keyword vector. To generate the keyword dictionary from users’ review, three steps are to be performed.

Step I Purifying. Since the reviews are user-generated, they usually contain spelling errors and non-recognizable words. It is necessary to make sure the dictionary doesn’t contain invalid words. In addition, numbers and punctuations are also excluded.

Step II Filtering. Not all the words in the reviews are useful, for example, prepositions and conjunctions are meaningless. In fact we are only interested in nouns which describe restaurant features and adjectives which might imply users’ feelings. Therefore, all the meaningless words are to be filtered out.

Step III Stemming. Removal of common word suffixes is done to decrease the size of dictionary. For instance, “apples” would be recognized as “apple” and “children” would be stored as “child” in the dictionary.

The three steps are completed by using the _WordNet_ [37] and _JWNL_ APIs [38]. The _WordNet_ is a large lexical database of English developed by Princeton University. Nouns,
verbs, adjectives and adverbs are grouped into sets of cognitive synonyms, each expressing a
distinct concept. JWNL is a Java API for accessing the WordNet relational dictionary. The
final keyword dictionary generated from users’ review has 12,990 words in total, including
all appeared nouns and adjectives.

After the keyword dictionary is generated, we created a weighted keyword vector for
each restaurant. The vector for a restaurant contains all 12,990 words in the dictionary, and
the weight for each keyword in the vector is the number of times the keyword has appeared
in the reviews of the restaurant. An example of a weighted keyword vector for a given
restaurant is shown in Figure 4.1. The keyword vectors are stored in files. In summary, the
representation of a restaurant entity is shown in Figure 4.2.
Table 4.2: Representation of Restaurant Entity

<table>
<thead>
<tr>
<th>rid</th>
<th>name</th>
<th>category</th>
<th>latitude</th>
<th>longitude</th>
<th>rating</th>
<th>review_count</th>
<th>url</th>
</tr>
</thead>
<tbody>
<tr>
<td>apgujung-vancouver</td>
<td>Apgujung</td>
<td>korean;</td>
<td>49.28989</td>
<td>123.13300</td>
<td>4</td>
<td>23</td>
<td><a href="http://www.yelp.ca/biz/apgujung-vancouver">http://www.yelp.ca/biz/apgujung-vancouver</a></td>
</tr>
<tr>
<td>guu-vancouver-2</td>
<td>Guu</td>
<td>japanese;</td>
<td>49.28422</td>
<td>123.12547</td>
<td>4.5</td>
<td>140</td>
<td><a href="http://www.yelp.ca/biz/guu-vancouver-2">http://www.yelp.ca/biz/guu-vancouver-2</a></td>
</tr>
<tr>
<td>au-petite-cafe-vancouver</td>
<td>Au Petite Cafe</td>
<td>vietnamese;</td>
<td>49.24123</td>
<td>123.10165</td>
<td>4</td>
<td>40</td>
<td><a href="http://www.yelp.ca/biz/au-petite-cafe-vancouver">http://www.yelp.ca/biz/au-petite-cafe-vancouver</a></td>
</tr>
</tbody>
</table>

Table 4.3: Representation of Restaurant Review

<table>
<thead>
<tr>
<th>ID</th>
<th>rid</th>
<th>rev_rating</th>
<th>rev_date</th>
<th>rev_context</th>
</tr>
</thead>
<tbody>
<tr>
<td>12001</td>
<td>acme-cafe-vancouver-3</td>
<td>5</td>
<td>5/19/2010</td>
<td>This place is so cute!! I loved the atmosphere…</td>
</tr>
<tr>
<td>12002</td>
<td>au-petite-cafe-vancouver</td>
<td>4</td>
<td>1/25/2011</td>
<td>Sandwiches! One of the best because of their very fresh and limited bread…</td>
</tr>
<tr>
<td>12003</td>
<td>green-lemongrass-richmond</td>
<td>5</td>
<td>11/13/2010</td>
<td>Good Vietnamese place with a really clean and beautiful menu. The salad roll was …</td>
</tr>
</tbody>
</table>

Figure 4.1: An Example of Weighted Keyword Vector
4.2 User Interest Model

As discussed in Chapter 2, a keyword-based user interest model is able to make good use of unstructured data while a hierarchical model is more powerful to deal with semantic issues. In this section a weighted tree, attached by a set of weighted keyword vectors, is proposed based on the user’s dining history and representation of restaurants.

4.2.1 Dining History Tree

For each user, a tree model that represents the user’s historical activities is firstly built up before his interest tree is created. In fact, it is the process of refactoring the record of user historical dining activities from a flat structure a tree structure. An example of dining history tree can be seen in Figure 4.3.
The root node of the tree stores user ID. The nodes at second level are category nodes, representing the categories of the restaurants that the user went before. Each node has two fields, ‘#Times’, which is the number of times that user went to the restaurants that belong to the category, and ‘Ratings’, which is the total ratings that the user gave to restaurants in the category. The leaf nodes are restaurants nodes which store the restaurants where the user had his meals in the past. Note that the summation of #Times and Ratings of all children nodes gives the value of corresponding fields in their parent node.

### 4.2.2 Interest Tree Representation

The user interest tree is an extension of dining history tree, as shown in Figure 4.4. Each node is attached a weighted keyword vector, which describes the features of the corresponding node. For example, the keyword vector of a restaurant node describes the feature of the restaurant, the keyword vector for a category node captures the features for all
restaurants under this category, and the keyword vector for root node is a description of all features that are preferred by the user. In addition, for both category nodes and restaurant nodes, an underlined field ‘Weight’ is added, which indicates how likely the user will prefer the category of a specific restaurant.

Figure 4.4: An Example of User Interest Tree

4.2.3 Computation of Interest Tree

The computation of user interest tree consists of two parts, the computation for keyword vectors and the computation for weight fields. They all follow a bottom-up way as described below.

To compute keyword vectors, firstly, for each restaurant node, retrieve its keyword vector from keyword vector file generated in section 4.1.3, and attach it to the node. Then
iteratively compute the keyword vector for the parent node by summing up all keyword vectors of the children nodes. User keyword vector is created at last. Denote the keyword vector of a node p as $p_{kv}$, the computation can be formalized as

$$p_{kv} = \sum_i c_{kv}$$

where $c_i$ is a child of node $p$.

To compute weight fields, let $n_p^k$ denote the node $p$ at level $k$, where $k = 0, 1, 2$, $W_p^k$ denote the weight of $n_p^k$, and $R_p^k$ denote the ratings of $n_p^k$, the computation of the weight of node $n_p^k$ can be expressed as

$$W_p^k = \frac{R_p^k}{\sum_j R_j^k}$$

### 4.2.4 Grey Node

The user interests tree does not only show what the user likes but also reflect what the user probably dislikes. For example, Figure 4.4 shows that user has gone to Chinese restaurants for only one time with a poor rating. It probably implies the restaurant, or even the type of food is of no interest to the user. Such a node is defined as ‘Grey Node’.

A bottom-up method is used to discover all the grey nodes. Firstly, for each restaurant node, if the values of both of its #Times and Ratings fields are below average value of the nodes at same level, then the node is marked as grey node. Then if all the restaurant nodes are grey nodes, the category node is also marked grey. Any restaurant that matches a grey node or is similar to a grey node shouldn’t appear in the recommendation results.
4.3 Personalized Recommendation

The recommendation problem can be generally formalized as follows [39]. Let $U$ be the set of users and $P$ be the set of items to be recommended. A score function $SF$ is defined to measure the score of item $p$ in the perspective of user $u$, i.e., $SF: U \times P \rightarrow R$, where $R$ is a totally ordered set. To make a personalized recommendation for a given $u \in U$, is to choose such item that maximizes the inferred score. Formally, we have

$$\forall u \in U, p_u = \arg \max_{p \in P} SF(u, p)$$

4.3.1 Score Function

In the case of restaurant recommendation, the restaurants to be recommended is called target restaurants. The recommender system calculates a score for every target restaurant and recommends the ones with the higher scores. Given a target restaurant to the user, the score that the restaurant would get in the user’s perspective consists of two parts, which are the category match score and the keyword vector match score. Formally, let $r$ be the target restaurant, $u$ be the user, $S_{r,c}^u$, $S_{r,k}^u$ and $S_r^u$ be the final score of the restaurant, the category match score and the keyword vector match score respectively, it gives that

$$S_r^u = \alpha * S_{r,c}^u + (1 - \alpha) * S_{r,k}^u$$

where $\alpha \in (0, 1)$ is the lubricating factor.
4.3.2 Category Match Score

For a target restaurant $r$, if the category of $r$ matches one category node in user’s interest tree, then the category match score is simply given by the weight of the category node, i.e., $S^u_{r,c} = W_c$, where $W_c$ is the weight of the matched category node. Here the word ‘match’ means that the category of the target restaurant is the same as the one of a category node in user’s interest tree. Note that a restaurant may have multiple categories that match multiple nodes in interest tree, resulting more than one category match scores. The strategy is to choose the highest match score as the final category match score, i.e.,

$$S^u_{r,c} = \max\{W_c \mid c \in \text{set of matched category nodes}\}$$

For target restaurant that doesn’t match any category node, the category match score gives zero.

4.3.3 Keyword Vector Match Score

For a target restaurant $r$ that doesn’t appear in user’s dining history tree, if the category of $r$ matches one category node in user’s interest tree, then the keyword vector match score is calculated by the cosine of the angle between the keyword vector of $r$ and the keyword vector of the matched node. If no category node matched, then the keyword vector match score is calculated by the cosine of the angle between the keyword vector of $r$ and the keyword vector of the root node.
Formally, let $t_{kv}$ and $c_{kv}$ be the keyword vector of $r$ and matched category node $c$ if any, respectively and let $r_{kv}$ be the keyword vector of root node, the keyword vector match score is given by

$$S_{r,k}^u = \begin{cases} 
\frac{t_{kv} \cdot c_{kv}}{\|t_{kv}\| \times \|c_{kv}\|}, & \text{if category node matched} \\
\frac{t_{kv} \cdot r_{kv}}{\|t_{kv}\| \times \|r_{kv}\|}, & \text{otherwise}
\end{cases}$$

4.3.4 Recommendation Algorithm

The input of the recommender system are user’s location $L$, which could either be an address or a coordinate, and a user’s preferred search range $R$, for example, 3 kilometers. The output is an ordered list of 10 restaurants that have the highest scores. The algorithm for recommending new restaurants to the user is described in Table 4.4.

<table>
<thead>
<tr>
<th>Recommendation Algorithm:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Generate a candidate set $C$ that contains all restaurants at within range of $R$ location $L$;</td>
</tr>
<tr>
<td>2. Initialize result set $RS = {}$;</td>
</tr>
<tr>
<td>3. For each restaurant $r$ in set $C$, find the matched category node $Nc$ in user interest tree with the highest weight:</td>
</tr>
<tr>
<td>a) If $Nc$ is a grey node, continue;</td>
</tr>
<tr>
<td>b) If $Nc$ exists, sum up the category match score and keyword vector match score as the final score $s$, then add $r$ with $s$ to $RS$;</td>
</tr>
<tr>
<td>c) If $Nc$ is null, calculate keyword vector match score using root node of interest tree as the final score $s$, then add $r$ with $s$ to $RS$;</td>
</tr>
<tr>
<td>4. Sort $RS$ with respect to final score $s$ of each entry $r$, and return the top 10 entries.</td>
</tr>
</tbody>
</table>
4.4 Summary

This chapter is the core work of the thesis. The entity representation is first investigated, based on which a hybrid user interest model is proposed. The advantage of this model is that it makes good use of unstructured data by having each node in the model attached a keyword vector, without losing the inherent hierarchical relationship between nodes. The interest mode is also able to discover what the user doesn’t like. In addition, a content-based personalized recommendation algorithm is suggested based on user interest model. The recommender system takes both of the restaurant style which is reflected by the weight of category node, and the features of restaurant which are represented by keyword vectors, into consideration.

A few of known limitations of the recommendation algorithm are, first, it is purely content-based, which implies that only the restaurants that are similar to the ones that user went before can appear in the recommended result; second, the algorithm wouldn’t work well when the user doesn’t have enough number of historical dining activities.

More discussions regarding to the validation of keyword vectors, the configurations of parameters and the effectiveness of the algorithm can be found in Chapter 6.
Chapter 5: LePlaza Platform – Visualization

LePlaza provides a user interface that visualizes user’s dining history, interests model and recommendation results. The aim of the visualization UI is to offer the user a straightforward view of his historical dining activities, a detailed analysis of his dining interests using charts and graphs, and an interactive manner of making a recommendation.

5.1 Dining History

User can view his historical dining activities by entering My Dining Activity page. Figure 5.1 is a snap of dining history visualization.

![My Dining Activities in past](image)

Figure 5.1: Dining History Visualization

By default, user’s dining activities in the last 3 months will be marked on the map, with different icons representing different types of food. The larger the icon is, the more
numbers of times the user has been there before. By dragging the slide bar above the map, user is allowed to choose to display his dining activities from last year to last two weeks. The detail of a dining activity will pop up when clicking any mark on the map, which includes the restaurant information such as ratings and address, and dining date.

On the right side is a list showing all the dining activities that are currently displayed on the map. The list can be sorted by number of times or the user’s rating for the dining activity.

5.2 Preference Analysis

The user’s interest model is visualized in My Preference page. In the preference summary page as shown in Figure 5.2, a pie chart is used to show the proportion that each type of food takes in the user’s all dining activities. A bar chart on the right side shows the user’s monthly dining activities, in terms of number of times. These charts would help the user to have a better understanding of his own preferences and dining behaviors. A text summary is provided at the bottom of the page, including the total number of dining activities, the average rating and cost, the most preferred food category and so on.

In the preference detail page, the pie chart is broken into separate parts, each of which representing a food category, with different sizes meaning different proportion in dining activities as shown in Figure 5.3. By double clicking any food category, the user can see a bar chart on the right showing the monthly dining activities within in one year of the selected
food category, with a text summary including the average rating, cost, and some highlighted features of the category.

Figure 5.2: Summary Page of Dining History

Figure 5.3: Detail Page of Dining History
This page also allows the user to play with his preference by dragging and dropping any food category to the pink circle in the middle. A window for recommendation will show up when some category is dropped in the drop area, and user can then be recommend with restaurants that serves the selected types of food within any specific range from user’s current location. Up to 10 recommended restaurants will be ranked and displayed on the map on right side, as shown in Figure 5.4.

Figure 5.4: Customized Recommendation Result

5.3 Recommendation Results

In My Recommendations page, a map is firstly displayed with the user’s current location marked in the middle, which is the default location where restaurants nearby will be recommended. However, user is free to change the location by dragging and dropping the
marker on the map. A recommendation event is fired and the result is ranked and displayed whenever the location is changed. The search range can be specified before moving the location marker and a list of recommended restaurants will be displayed on the right afterwards. Figure 5.5 gives an example of recommendation result.

![Map with recommended restaurants](image)

**Figure 5.5: Location Based Recommendation Result**

<table>
<thead>
<tr>
<th>No.</th>
<th>Restaurant Name</th>
<th>Cuisine</th>
<th>Cost Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shii</td>
<td>Sushi Bar; Japanese</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>Ziqiang Sushi</td>
<td>Sushi Bar; Japanese</td>
<td>$6</td>
</tr>
<tr>
<td>3</td>
<td>Taipan Restaurant</td>
<td>Japanese; Sushi Bar</td>
<td>$5555</td>
</tr>
<tr>
<td>4</td>
<td>Sushi</td>
<td>Japanese; Sushi Bar</td>
<td>$5555</td>
</tr>
<tr>
<td>5</td>
<td>Taisho Japanese Dining</td>
<td>Japanese; Sushi Bar</td>
<td>$5555</td>
</tr>
<tr>
<td>6</td>
<td>You Japanese Taiki</td>
<td>Asian Fusion; Japanese</td>
<td>$5555</td>
</tr>
<tr>
<td>7</td>
<td>Maruha Japanese BBQ Dining</td>
<td>Asian Fusion; Japanese</td>
<td>$5555</td>
</tr>
<tr>
<td>8</td>
<td>G O Ramen</td>
<td>Sushi; Japanese</td>
<td>$</td>
</tr>
<tr>
<td>9</td>
<td>Ajiyon Hanabi</td>
<td>Food Stands; Japanese</td>
<td>$</td>
</tr>
<tr>
<td>10</td>
<td>Shoyunen Noodle House</td>
<td>Food Stands; Japanese</td>
<td>$1</td>
</tr>
</tbody>
</table>
Chapter 6: Evaluation

In this chapter, the effectiveness of user interest model and personalized recommendation will be evaluated. The parameter used in recommendation module is determined by experiments. A user study is conducted and feedback collected from participants is provided as a proof of system feasibility.

6.1 Effectiveness of Keyword Vector

Recall that in the personalized recommendation discussed in section 4.3.1, the final score for a target restaurant to be recommended to a specific user is given by

\[ S_r^u = \alpha S_{r,c}^u + (1 - \alpha) S_{r,k}^u \]

where the first part \( S_{r,c}^u \) is the category match score and the second part \( S_{r,k}^u \) is the keyword match score. \( S_{r,c}^u \) is just the weight of the matched category node in user interest tree, which is pre-computed and stored and can be easily read from the tree when making a recommendation. In the second part, \( S_{r,k}^u \) measures the similarity between the target restaurant and the matched node in user interest tree, which is also the cosine value of the angle between the two corresponding keyword vectors. However, the calculation of \( S_{r,k}^u \), on the other hand, is not that straightforward because it is not a pre-computed value that can be viewed or retrieved somewhere.

Therefore, in order to evaluate the effectiveness of using the extracted keyword vectors as a measurement for the similarity between two restaurants and to have an intuitive
understanding of quantity of the similarity, we defined intra-category similarity and inter-category similarity respectively.

6.1.1 Intra-category Similarity

Firstly, the intra-category similarity for a given category is defined as the average similarity between any of the two restaurants within the category. Formally, let \( C_i \) be the \( i \)th category, \( R_i \) is the set of restaurants within the category, and \( S_i \) is the intra-category similarity for \( C_i \), then \( S_i \) is given by

\[
S_i = \frac{\sum_{u \in R_i, v \in R_i, u \neq v} \text{sim}(u, v)}{N_i}
\]

where \( u \) and \( v \) are restaurants within category \( C_i \), \( \text{sim}(u, v) \) is the similarity between \( u \) and \( v \), and \( N_i \) is the number of \((u, v)\) pairs in \( C_i \).

We can also define \( S_{i, \text{max}} \) and \( S_{i, \text{min}} \) in the same way,

\[
S_{i, \text{max}} = \max\{\text{sim}(u, v) \mid u \in R_i, v \in R_i, u \neq v\}
\]

\[
S_{i, \text{min}} = \min\{\text{sim}(u, v) \mid u \in R_i, v \in R_i, u \neq v\}
\]

where \( S_{i, \text{max}} \) means the maximum similarity between two restaurants within category \( C_i \) and \( S_{i, \text{max}} \) means the minimum similarity within \( C_i \). Table 6.1 shows the \( S_{i, \text{max}}, S_{i, \text{min}} \) and \( S_i \) for the largest ten categories.
<table>
<thead>
<tr>
<th>Category</th>
<th>Number of Rests</th>
<th>$S_{l,\text{max}}$</th>
<th>$S_{l,\text{min}}$</th>
<th>$S_{l}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotdogs</td>
<td>45</td>
<td>0.9257702</td>
<td>0</td>
<td>0.0950225</td>
</tr>
<tr>
<td>Chinese</td>
<td>42</td>
<td>0.7988825</td>
<td>0</td>
<td>0.2257978</td>
</tr>
<tr>
<td>Indpak</td>
<td>39</td>
<td>0.7942415</td>
<td>0</td>
<td>0.3264948</td>
</tr>
<tr>
<td>Japanese</td>
<td>38</td>
<td>0.8487532</td>
<td>0.04476907</td>
<td>0.3394094</td>
</tr>
<tr>
<td>Seafood</td>
<td>36</td>
<td>0.8783334</td>
<td>0</td>
<td>0.2552037</td>
</tr>
<tr>
<td>breakfast_brunch</td>
<td>32</td>
<td>0.9234331</td>
<td>0.01918471</td>
<td>0.250193</td>
</tr>
<tr>
<td>Burgers</td>
<td>32</td>
<td>0.7062084</td>
<td>0</td>
<td>0.1843502</td>
</tr>
<tr>
<td>Sandwiches</td>
<td>32</td>
<td>0.7303357</td>
<td>0</td>
<td>0.1723489</td>
</tr>
<tr>
<td>Newamerican</td>
<td>28</td>
<td>0.6733683</td>
<td>0.03160561</td>
<td>0.3013634</td>
</tr>
<tr>
<td>Mideastern</td>
<td>25</td>
<td>0.6962025</td>
<td>0.00847397</td>
<td>0.2909511</td>
</tr>
</tbody>
</table>

The table shows that the similarities between restaurants are nearly distributed in the range of (0, 1) with a mean around 0.3. The $S_{l,\text{max}}$ column shows that under each category, there exists some restaurants that are pretty similar to each other. More results show that the maximum similarity among all restaurants is 0.9257702, which is given by two hotdogs restaurants that both sell japa-dogs, ‘japa-dog-vancouver’ and ‘japa-dog-vancouver-2’, located at different streets in Vancouver downtown. This makes sense because it is highly possible that a japa-dog lover will get recommended with other japa-dogs nearby when he is hanging around.

### 6.1.2 Inter-category Similarity
Secondly, the inter-category similarity is defined as the similarity between two categories. For a given pair of categories, the inter-category similarity is the average similarity between any of the two restaurants, each of which falls into one category. Formally, let \( C_i \) and \( C_j \) be two given categories, and \( R_i \) and \( R_j \) are the sets of restaurants that belong two \( C_i \) and \( C_j \) respectively. Let \( S_{i,j} \) be the inter-category similarity for \( C_i \) and \( C_j \), which is given by,

\[
S_{i,j} = \frac{\sum_{u \in R_i, v \in R_j} \text{sim}(u, v)}{N_{i,j}}
\]

where \( u \) and \( v \) are restaurants within category \( C_i \) and \( C_j \) respectively, and \( N_{i,j} \) is the number of \((u, v)\) pairs. Table 6.2 gives the inter-category similarities for 6 selected typical food categories: Greek, Italian, French, Chinese, Malaysian and Thai food.

**Table 6.2: Inter-category Similarities**

<table>
<thead>
<tr>
<th></th>
<th>Greek</th>
<th>Italian</th>
<th>French</th>
<th>Chinese</th>
<th>Malaysian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Italian</td>
<td>0.20465</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>French</td>
<td>0.211486</td>
<td>0.29955</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Chinese</td>
<td>0.14474</td>
<td>0.168456</td>
<td>0.184176</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malaysian</td>
<td>0.159773</td>
<td>0.162888</td>
<td>0.184505</td>
<td>0.235711</td>
<td></td>
</tr>
<tr>
<td>Thai</td>
<td>0.141429</td>
<td>0.153315</td>
<td>0.173946</td>
<td>0.255981</td>
<td>0.278918</td>
</tr>
</tbody>
</table>

The average of the whole table is 0.1973. It can be observed that the values in upper-left triangle and the bottom-right triangle are all above average, and the rest are below average. The result corresponds well with the fact that Greece, France and Italy are geographically close to each other and they have pretty similar food style. While the other
group of country, China, Malaysia and Thailand has the same conclusion, it can be seen that there is not too much in common for categories such as Thai and Greek, which is also the truth as we know.

Not surprisingly, the conclusion draw from the two similarity experiments applies well with people’s common sense. It does demonstrate that the keyword vectors that are extracted from users’ reviews work well and they are effective measurements for making recommendations.

6.2 Determine Lubricating Factor

One parameter to be determined is the lubricating factor $\alpha$ in the recommendation score function

$$S^u_r = \alpha S^u_{r,c} + (1 - \alpha) S^u_{r,k}$$

The factor is used to keep the balance of the category match score $S^u_{r,c}$ and the keyword vector match score $S^u_{r,k}$ that a target restaurant would get. The intuition is that if $\alpha$ is too large (close to 1), then the category match score will dominate the final score, resulting in a very limited coverage of the recommendation result as the most of recommended restaurants will probably fall only into a very few of categories preferred by the user. On the other hand, if $\alpha$ is too small (close to 0), the user would probably be disappointed to see he is recommended with a lot of unfamiliar cuisines that he hardly tried before, which doesn’t reflect the user’s preferences on food styles well.

6.2.1 Measurement D
To determine a proper value of $\alpha$, some measurements need to be defined first. We define the user interest distribution vector for a given list of food categories as the vector containing in sequence the percentage of each category in the user’s historical dining activities, denoted by $V_u^I$, where $I$ is list of categories. For example, if a user had 10 dining activities in the past, with 5 times in Japanese food, 3 times French food and 2 times pizza, then the user interest distribution vector for category list (Japanese, French, pizza) is $V_u^I = (0.5, 0.3, 0.2)$. Similarly, we also define the global distribution for a given list of food category as the percentage of each food category of all restaurants in database, denoted by $V_g^I$. For example, in all 816 restaurants we have, there are 45 Japanese, 20 French and 23 pizza restaurants. Thus the global distribution vector for category list (Japanese, French, pizza) is $V_g^I = (45/816, 20/816, 23/816)$.

In the same way, we can define the recommended distribution vector for a given list of food category in the recommendation results, which is the percentage of each category in recommendation results, denoted by $V_{rec}^I$. Let’s still take the list (Japanese, French, pizza) as an example. If the recommended distribution vector is very close to the user interest distribution vector $V_u^I$, this implies the factor $\alpha$ is too large and the food categories dominate recommendation results. If the recommended distribution vector is very close to the global distribution vector $V_g^I$, this implies the factor $\alpha$ is too small and the user’s preferences on food category are not reflected by recommendation results.
Therefore, it is reasonable to keep $V_{rec}^I$ in between $V_u^I$ and $V_g^I$, neither too far from $V_u^I$ nor too far from $V_g^I$. The distance between the vectors are defined as the measurement $D$ where

$$D = \left\| V_{rec}^I - V_u^I \right\| + \left\| V_{rec}^I - V_g^I \right\|$$

where $\|\|$ is the Euclidean distance between two vectors. It is obvious that the recommendation results vary with $\alpha$, resulting in the change of $V_{rec}^I$ and thus the change of $D$. The purpose is then to select the factor $\alpha$ such that

$$\alpha = \arg\min(D)$$

### 6.2.2 Experiment Result

In the experiment, we created 6 fake users, and each of the users has 40 automatically generated dining activities that are nearly evenly distributed in 8 randomly picked food categories. For instance, fake user 1 has 40 dining activities for food categories (hotdogs, sandwiches, greek, foodstands, sushi, dimsum, creperies, ethnicmarkets) with a user interest distribution vector $V_u^I = (0.15, 0.15, 0.175, 0.125, 0.125, 0.125, 0.125, 0.125)$. The corresponding global distribution vector $V_g^I$ is

$$V_g^I = (0.0551, 0.0392, 0.0281, 0.0270, 0.0257, 0.0245, 0.0196, 0.0147).$$

With $\alpha$ varying from 0.1 to 0.8, the recommendation algorithm is run against 6 fake users and for each of the users 100 restaurants are recommended among all 816 restaurants. A recommended distribution vector $V_{rec}^I$ is then calculated from the recommendation result for
each user and finally the average distance $D$ is computed. Figure 6.1 shows how the measurement $D$ varies with $\alpha$.

![Average Distance (D)](image)

Figure 6.1: Measurement D Varies with $\alpha$.

The figure shows that the curve of average distance is concave which has the minimum value when $\alpha = 0.3$, and $\alpha = 0.4$ gives the similar result. Therefore, we fix $\alpha$ to 0.3 in recommendation score function for user evaluations discussed in next section.

6.3 User Study

To evaluate the effectiveness of the personalized recommender system a user test with 8 participants was conducted. Each user was asked to give 30-50 historical dining activities according to his personal preference, including the number of times and ratings, in 100 randomly picked restaurants. Based on the provided dining activities, a list containing the top 10 restaurants among another randomly selected 100 restaurants is recommended to
the user. Feedback of the recommendation result from the participants are collected and studied.

6.3.1 Metrics

Various metrics have been used to evaluate the performance of recommender systems. We choose the most common two metrics in this thesis, which are precision and recall.

Precision $P$ is defined as the ratio of the number of relevant items to the total number of items in the recommendation result, and recall $R$ is defined as the ratio of the number of relevant items to the number of all relevant items available, as shown in formula [40] below.

$$P = \frac{N_{rs}}{N_s}, \quad R = \frac{N_{rs}}{N_r}$$

where $N_s$ is the total number of items in recommendation result, $N_r$ is the total number of relevant items, and $N_{rs}$ is the number of relevant items that appear in recommendation result. In our test case, $N_s$ is always equal to 10, $N_{rs}$ is the number of the recommended restaurants that the user is satisfied with, and $N_r$ is the total number of restaurants that are interesting to the user and may or may not be included in the recommendation result.

6.3.2 Experiment Results

We compared our personalized recommendation module in LePlaza system with a simple traditional one that makes recommendations according to the average ratings and number of reviews generated by all users. Table 6.3 and Figure 6.2 show the precision of the
two systems from all 8 participants, who are denoted by \( P0 \) to \( P7 \). Table 6.4 and Figure 6.3 show the recall values.

### Table 6.3: Precision Comparison

<table>
<thead>
<tr>
<th></th>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LePlaza</td>
<td>0.8</td>
<td>0.8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>0.9</td>
<td>1</td>
</tr>
<tr>
<td>Traditional</td>
<td>0.5</td>
<td>0.7</td>
<td>0.6</td>
<td>0.8</td>
<td>0.8</td>
<td>0.7</td>
<td>0.7</td>
<td>0.4</td>
</tr>
</tbody>
</table>

![Precision Comparison](image)

**Figure 6.2: Precision Comparison**

### Table 6.4: Recall Comparison

<table>
<thead>
<tr>
<th></th>
<th>P0</th>
<th>P1</th>
<th>P2</th>
<th>P3</th>
<th>P4</th>
<th>P5</th>
<th>P6</th>
<th>P7</th>
</tr>
</thead>
<tbody>
<tr>
<td>LePlaza</td>
<td>0.62</td>
<td>0.89</td>
<td>0.90</td>
<td>0.75</td>
<td>0.69</td>
<td>0.90</td>
<td>0.69</td>
<td>1</td>
</tr>
<tr>
<td>Traditional</td>
<td>0.33</td>
<td>0.58</td>
<td>0.50</td>
<td>0.82</td>
<td>0.57</td>
<td>0.64</td>
<td>0.58</td>
<td>0.33</td>
</tr>
</tbody>
</table>
Some observations can be made. Firstly, for almost all users, our personalized recommender system performs much better than the traditional one, which demonstrates that the proposed user interest model reflects users’ interests. Secondly, it can be seen from Figure 6.2 that the precision values of LePlaza for 8 users appear stable and they are all above 80%, which suggests that the LePlaza recommender system would be effective in real use. Thirdly, there are some obvious variations for recall metric in Figure 6.3. Even for LePlaza the best case is 100% while the worst case is only 69%. This is probably because in the experiment we hard-coded to recommend only the top 10 restaurants rather than all possible relevant restaurants.

In addition, 4 out of 8 participants complained that the first restaurant that appeared in the recommended results was not satisfying when using the simple traditional recommender system, while this number of people decreased to 1 for our LePlaza system.
Chapter 7: Conclusion

People often make efforts to enrich their social lives by seeking answers to questions like “Are there any exciting movies coming out next week that I can enjoy with my friends?” “Is there a party taking place in my neighborhood?” or “Which restaurant nearby fits my taste best?”. To answer such questions, LePlaza, which is a location-based and event-centered social networking platform, is presented in this thesis.

7.1 Contributions

Firstly, the whole framework and fundamental functions of LePlaza system are designed and implemented, on which users can create, browse, search and manage variety types of social events. Moreover, group operations are supported so that the user can manage friends and events in group with ease and flexibility.

Secondly, as a particular case of events, restaurants and user’s dining activities are studied, based on which a content-based hybrid user interest model is presented. Experimental results show that the proposed model can capture user’s dining preference effectively.

Thirdly, a personalized recommendation algorithm is proposed. It is location-based and can recommend restaurants that meet the user’s taste within any given area according to the user interest model. A user study shows that the performance of the recommendation is satisfying with an average accuracy over 80%.
Finally, both user interest model and recommendation result are visualized in an interactive manner, aiming to give the user a better understanding of his preference and make him feel in control of his experience.

7.2 Future Work

- Generalized user interest model. The current user interest model is built on top of the dining events only. It would be interesting to apply the interest model to other types of events. For example, a real estate application can use this model to give suggestions to people who are seeking for houses, according to location information and house descriptions. Other applications in which the model applies include shopping guide, academic conference recommendation, etc. In all, the model is expected to be generally applicable to many other applications since the tree structure in the model can still be remained and new nodes for other types of event could be easily attached to the tree.

- Group interest model. To come up with a way that combines individual interest models to get a model for a group of people would be useful when an agreement or a decision has to be made among group members. For example, what event will be probably enjoyed by most of the people in a group when they are going to have some group activities?
• Mobile side support. Location based services are often associated with mobile applications. With a LePlaza client installed on mobile phone, users will be able to find interesting events in real time at any places.
Bibliography


[17] Liu, D., “The Design and Research of User Interest Model in Personalized Search


Appendix: Original Data of Restaurant Recommendation User Feedback

Eight participants were asked to fill “Bad?” column with number “1” or “2” according to the recommended restaurants. Number “1” represents that the user is not happy with the restaurant, and number “2” means the user is not satisfied with the rank of the restaurant in the result.

**P0.xls**

<table>
<thead>
<tr>
<th>Recommended Restaurants:</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| No. | Restaurant | Cuisine; Highlights | Price | Bad?
| 1  | Sulka | Tapas; Small Plates; Japanese; baked tapioca pudding; Hong Kong; deep-fried tofu | $5 | 2
| 2  | Kalamata Greek Taverna | Greek; Mediterranean | $5 |
| 3  | Kirin Seafood Restaurant | Dim Sum; | $5 |
| 4  | Tropika On Cumbie | Indonesian; Malaysian; Singaporean; Deep fried Tofu, Malay; always busy | $5 |
| 5  | Banana Leaf | Malaysian; | $5 |
| 6  | Pinto Estiatorio | Greek; | $5 |

| Cru Restaurant | French; course prix fixe menu; goat cheese cab; Beef Tenderloin | $5 |
| Shiro | Sushi Bars; Japanese; Lunch box G; wasabi; sashimi | $5 |
| 9  | 3G Vegetarian Restaurant | Vegetarian; | $5 |
| 10 | Roman's Burger Bar | Burgers; | $5 |

**P1.xls**

<table>
<thead>
<tr>
<th>Recommended Restaurants:</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
</table>
| No. | Restaurant | Cuisine; Highlights | Price | Bad?
| 1  | No.1 Beef Noodle | Taiwanese; | $ | 0
| 2  | Green Basil Thai Restaurant | Thai; Lunch specials; tom yum soup; curry | $ | 2
| 3  | Wang's Shunghai Cuisine | Chinese; XLR; dumplings; dim | $ |
| 4  | Huiji Noodle Specialist | Food Stands; Ethnic Food; Chinese; | 0
| 5  | No 1 Beef Noodle House | Taiwanese; | 0
| 6  | Chilli Pepper House | Asian Fusion; Chinese; Indian; | $ |
| 7  | Angkor Restaurant | Cambodian; | $ |
| 8  | Western Lake Chinese Restaurant | Chinese; dim-sum; make a reservation; very reasonable | $ |
| 9  | Agra Tandoori Restaurant | Indian; Caterers; | $ |
| 10 | Mini Garden Restaurant | Malaysian; Chinese; | $ |

**P2.xls**
### Recommended Restaurants:

<table>
<thead>
<tr>
<th>No.</th>
<th>Restaurant</th>
<th>Cuisine</th>
<th>Highlights</th>
<th>Price</th>
<th>Bad?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sea Harbour Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$$$$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Top Gun J &amp; C Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Kirin Seafood Restaurant</td>
<td>Seafood;</td>
<td>dim sum; Mango pudding; Vancouver</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Fisherman's Terrace Seafood Restaurant</td>
<td>Seafood;</td>
<td>dim sum; Sun Sui Wah; shrimp dumplings</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Imperial Court Beijing Cuisine</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Mui Garden Restaurant</td>
<td>Malaysian;Chinese;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Northern Delicacy</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Tropika At Aberdeen</td>
<td>Indonesian;Malaysian;Singapore;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Kung Bok Palace</td>
<td>Korean;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Red Star Seafood Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$$$$</td>
<td></td>
</tr>
</tbody>
</table>

### P3.xls

<table>
<thead>
<tr>
<th>No.</th>
<th>Restaurant</th>
<th>Cuisine</th>
<th>Highlights</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Golden Swan Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>Zipang Sushi</td>
<td>Sushi Bars;Japanese;</td>
<td>spicy cactus; sushi; takoyaki</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>Camy's Pizza</td>
<td>Pizza;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>Fraser Court Seafood Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>Supreme Pizza</td>
<td>Pizza;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>6</td>
<td>Kam Wah Loung</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>7</td>
<td>Sawasdee Thai Restaurant</td>
<td>Thai;</td>
<td>Coconut rice; red curry; papaya salad</td>
<td>$</td>
</tr>
<tr>
<td>8</td>
<td>Long's Noodle House</td>
<td>Chinese;</td>
<td>spicy beef noodle soup; drunken chicken; pea shoots</td>
<td>$</td>
</tr>
<tr>
<td>9</td>
<td>Bob Likes Thai Food</td>
<td>Thai;</td>
<td>Panang beef; papaya salad; swimming rum</td>
<td>$</td>
</tr>
<tr>
<td>10</td>
<td>Hawker's Delight Deli</td>
<td>Malaysian;</td>
<td>mee goreng; veggie fritters; Hainanese Chicken Rice</td>
<td>$</td>
</tr>
</tbody>
</table>

### P4.xls

<table>
<thead>
<tr>
<th>No.</th>
<th>Restaurant</th>
<th>Cuisine</th>
<th>Highlights</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sun Sui Wah Seafood Restaurant</td>
<td>Dim Sum;Seafood;</td>
<td>Vancouver; long long; Har Gow</td>
<td>$</td>
</tr>
<tr>
<td>2</td>
<td>Top Shanghai Cuisine Restaurant</td>
<td>Chinese;</td>
<td>Xiao long bao; dumplings; pan fried</td>
<td>$</td>
</tr>
<tr>
<td>3</td>
<td>Dinesty Chinese Restaurant</td>
<td>Chinese;</td>
<td>Xiao long bao; Din Tai Fung; onion pancake</td>
<td>$</td>
</tr>
<tr>
<td>4</td>
<td>Kirin Seafood Restaurant</td>
<td>Seafood;</td>
<td>dim sum; Mango pudding; Vancouver</td>
<td>$</td>
</tr>
<tr>
<td>5</td>
<td>Fisherman's Terrace Seafood Restaurant</td>
<td>Seafood;</td>
<td>dim sum; Sun Sui Wah; shrimp dumplings</td>
<td>$</td>
</tr>
<tr>
<td>6</td>
<td>Sea Harbour Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$$$$</td>
</tr>
<tr>
<td>7</td>
<td>Taco Luís</td>
<td>Mexican;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>8</td>
<td>Top Gun J &amp; C Restaurant</td>
<td>Dim Sum;</td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>9</td>
<td>Chef Hung Taiwanese Beef Noodle</td>
<td>Taiwanese;</td>
<td>Braised beef shank; Aberdeen Center; fire chili</td>
<td>$</td>
</tr>
<tr>
<td>10</td>
<td>Tropika At Aberdeen</td>
<td>Indonesian;Malaysian;Singapore;</td>
<td></td>
<td>$</td>
</tr>
</tbody>
</table>

### P5.xls
### P6.xlsx

**Recommended Restaurants:**

<table>
<thead>
<tr>
<th>No.</th>
<th>Restaurant</th>
<th>Cuisine</th>
<th>Highlights</th>
<th>Price</th>
<th>Bad?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>La Bretonnne the Original Creperie</td>
<td>Creperies;</td>
<td>crepe; buckwheat; savoury</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Kodoya Japanese Restaurant</td>
<td>Japanese; Sushi Bars;</td>
<td>snow white roll; spicy tuna; service was super fast</td>
<td>$$</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Keg Steakhouse &amp; Bar</td>
<td>Steakhouses;</td>
<td>prime rib; sweet potato fries; spinach salad</td>
<td>$$$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Vera's Burger Shack</td>
<td>Burgers;</td>
<td>can't beat Vera's meat; onion rings; poutine</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Cafe Crepe</td>
<td>Creperies; Breakfast &amp; Brunch;</td>
<td>nutella; sweet and savoury; banana</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Bon Crepe</td>
<td>Creperies;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Doolin's Irish Pub</td>
<td>Irish; Pubs;</td>
<td>pulled pork sandwich; good time; Guinness</td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>YEW restaurant + bar</td>
<td>American (New); Wine Bars;</td>
<td>Service is top notch; hotel; brunch</td>
<td>$$$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Romano's Pizza</td>
<td>Pizza;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Cafe Crepe</td>
<td>Creperies; Breakfast &amp; Brunch;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### P7.xlsx

**Recommended Restaurants:**

<table>
<thead>
<tr>
<th>No.</th>
<th>Restaurant</th>
<th>Cuisine</th>
<th>Highlights</th>
<th>Price</th>
<th>Bad?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cafe Crepe</td>
<td>Creperies; Breakfast &amp; Brunch;</td>
<td>Nutella; ham; Island</td>
<td>$</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Rocky Mountain Flatbread Company</td>
<td>Pizza;</td>
<td>fig &amp; goat cheese salad; organic; wood-fired course price fixe menu; goat cheese cake;</td>
<td>$$</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Cru Restaurant</td>
<td>French;</td>
<td>Beef Tenderloin</td>
<td>$$$</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Bistrot Bistro</td>
<td>French;</td>
<td>chocolate mousse; Beef bourguignon; goat cheese salad</td>
<td>$$$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Hell's Kitchen</td>
<td>Pizza; American (New);</td>
<td>Green Monster; Kits; chili sauce</td>
<td>$$</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Acropol Greek Restaurant</td>
<td>Greek;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Pinto Estiatorio</td>
<td>Greek;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Bistro Pastis</td>
<td>French;</td>
<td></td>
<td>$$$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Nando's Chicken</td>
<td>Chicken Winge;</td>
<td></td>
<td>$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Olympia Pizza</td>
<td>Greek; Pizza;</td>
<td>Lamb souvlaki; absolute; potatoes</td>
<td>$</td>
<td></td>
</tr>
</tbody>
</table>