Hindawi Publishing Corporation
Advances in Multimedia
Volume 2012, Article ID 987124, 9 pages
doi:10.1155/2012/987124

Research Article

Mining Local Specialties for Travelers by Leveraging Structured and Unstructured Data

Kai Jiang,[1] Like Liu,[2] Rong Xiao,[2] and Nenghai Yu[1]

[1] *MOE-Microsoft Key Laboratory of Multimedia Computing and Communication, University of Science and Technology of China, Anhui, Hefei 230027, China*
[2] *Microsoft Research Asia, Tower 2, No. 5 Dan Ling Street, Haidian District, Beijing 100080, China*

Correspondence should be addressed to Kai Jiang, kaijiang@mail.ustc.edu.cn

Received 10 May 2012; Accepted 26 June 2012

Academic Editor: Lei Wu

Recently, many local review websites such as Yelp are emerging, which have greatly facilitated people's daily life such as cuisine hunting. However they failed to meet travelers' demands because travelers are more concerned about a city's local specialties instead of the city's high ranked restaurants. To solve this problem, this paper presents a local specialty mining algorithm, which utilizes both the structured data from local review websites and the unstructured user-generated content (UGC) from community Q&A websites, and travelogues. The proposed algorithm extracts dish names from local review data to build a document for each city, and applies *tfidf* weighting algorithm on these documents to rank dishes. Dish-city correlations are calculated from unstructured UGC, and combined with the *tfidf* ranking score to discover local specialties. Finally, duplicates in the local specialty mining results are merged. A recommendation service is built to present local specialties to travelers, along with specialties' associated restaurants, Q&A threads, and travelogues. Experiments on a large data set show that the proposed algorithm can achieve a good performance, and compared to using local review data alone, leveraging unstructured UGC can boost the mining performance a lot, especially in large cities.

1. Introduction

The notion of *SoLoMo* (social local mobile) has induced an explosion of mobile technologies and applications. Under this trend, many local review social network services such as Yelp [1], Dianping [2], and Baidu Shenbian [3] are emerging. These websites enable users to explore, search, share and review local business entities, and indeed provide valuable information for people's daily life. Take cuisine hunting for instance, these applications may provide a great answer to the question "What are the fabulous restaurants nearby and what are the featured dishes in these restaurants?" That might satisfy local residents, but for a traveler, that's not enough. What makes travelers different from local residents is that instead of nearby restaurant and their featured dishes, a traveler is more concerned about the local specialties of the city. A local specialty means a dish is so special in some way that it seldom found in other cities. It may be the ingredients, flavor or cooking style that makes the dish special, and the local specialty often reveals the local culture and lifestyle. Thus, to experience local specialties is always an important task for travelers. Unfortunately, current local review services cannot meet travelers' demands well, because of the following.

(1) Current local review services tend to recommend restaurants of high rank to users, but a city's local specialties are not necessarily provided in high-ranked restaurants. So the user will not be able to discover the specialties.

(2) In some cities, especially large ones, restaurants' quantity can be very large and their varieties can be vast. In these cities, the local specialties may be overwhelming and cannot be found by travelers. For example, Baidu Shenbian contains 49903 restaurant pages in Beijing, and these restaurants cover almost all dish varieties in China, such as Chuan-Style, Xiang-Style, Lu-Style and even foreign dish varieties,

such as Italian-Style, Korean-Style, and Japanese-Style. It is very hard for a traveler to dig into this vast information to find out what is the specialty of Beijing.

To solve this problem, this paper propose a local specialty mining algorithm, which utilizes both the structured data from local review websites and the unstructured data from community Q&A websites and travelogues. We have noticed that many travelers may ask information about travel destinations on Q&A websites such as Yahoo! Answers [4], and after the trip travelers like to record their travel experience in travelogues. We believe the community Q&A data and travelogues can reveal valuable information about travel destinations, so these unstructured user-generated content (UGC) are adopted into our mining algorithm.

Our method first extracts dish names from restaurants' featured dishes information in the local review data. After that the dish names are filtered to remove trivial dish names, noises, and spams. After the dish filtering, a document is built for each city. The words in the document are dishes that are recommended by the city's restaurants. Then *tfidf*-weighting algorithm is applied to these documents to rank dishes. As for the unstructured UGC, locations are first extracted from Q&A threads and travelogues, and then the correlation of dishes and cities are calculated. The *tfidf*-ranking score generated from local review data and city-dish correlation score generated from unstructured UGC are combined to generate the final ranking score. Dishes with high-ranking score are considered as local specialties. Duplicate dishes in top local specialties are merged and reranked to form the final local specialty mining results. After the mining process, a recommendation service is built which can recommend local specialties to a traveler, and for each local specialty, its associated restaurants, Q&A threads, and travelogues are ranked and presented. Extensive experiments demonstrate that leveraging both structured local review data and unstructured UGC can achieve a good local specialty mining performance, thus the effectiveness of our method is proven.

The contributions of this paper are as follows. (1) To the best of our knowledge, this is the first paper addressing the novel problem of local specialty mining and is of particular interest for travelers. (2) This paper presented a method that leverages both structured local review data and unstructured UGC, which generates a good mining performance. (3) A recommendation service is built to recommend local specialties to travelers. The local specialties' associated restaurants, Q&A threads, and travelogues are also presented to travelers, so that these information can facilitate travelers' cuisine hunting.

The remainder of this paper is organized as follows. Section 2 reviews some related works. Section 3 formulates the local specialty mining problem, and gives an overview of the proposed mining algorithm. The local specialty mining algorithm is elaborated in Section 4, followed by the description of the recommendation service in Section 5. In Section 6, the experiment's settings are first introduced, and then the effectiveness of the proposed algorithm is evaluated, and finally the results are reported, followed by some discussions. Section 7 concludes the paper and presents some future research directions.

2. Related Work

There are some research efforts which are related to our work. Here we give these works a brief description from three directions.

UGC as Contextual Information. There are some works focused on landmark and tourism attraction mining for travel recommendation. These works adopted user-generated content such as blogs, user reviews, and user ratings, to serve as contextual information. Gao et al. [5] build a tourism recommendation service by mining landmarks from geotagged photos in photo sharing websites such as Flickr. In the landmark mining process, Yahoo Travel Guide [6] is adopted as a context information to decide whether the tags from photo sharing websites are travel related, and in the landmark ranking process, user-generated reviews and ratings in Yahoo Travel Guide is also brought into use. Ji et al. [7] harvest travel related photos and blogs from Windows Live Space [8], and associate the photos with extracted locations from the textual information such as photos' titles, tags, and blogs. After that, both the photos' visual information and the location hierarchy are used to rank and recommend attractions.

Travelogue Mining. There're many works which are dedicated to extract location related information by mining the large volume of user-generated travelogue [9–12]. Ye et al. [9] are focused on identifying a travelogue's theme location when there are multiple locations in a travelogue. [10, 12] apply a generative model to train location related local topic from travelogues, and extract locations' representative textual tags according to these local topics. Furthermore, these representative tags are used to retrieve related photos. Both representative tags and photos are clustered, ranked and, organized to give the user a better understanding about a specific location. In [11] Hao et al. not only generated locations' textual and visual summarizations as what they did in [10, 12], but also fully utilized the location's local topics trained from travelogues to perform the travel destination recommendation with respect to a user's query.

Local Dish Mining. The works which are most similar to us are [13, 14], which are dedicated to dish mining. [13] is focused on dish names extraction from restaurant reviews, and [14] pushes this work forward. Besides dish name extraction, [14] also tries to extract restaurant names from users' blogs and map them to a POI database, so that the extracted dish can be matched to restaurants in the POI database and can be deployed into a mobile map service. However, these works differ from us in the following (1). These works are focused on dish name extraction and restaurant name extraction, which are related to named entity extraction and recognition problem. But in our work, dish names are directly obtained from the local review websites, and our work is focused on the city's local specialty mining, which is related to ranking and recommendation problem. (2) The dishes extracted by [13, 14] are general dishes provided by

Figure 1: Example of local review websites. (a) Yelp, (b) Shenbian.

local restaurants which are more suitable for local residents, while dishes mined by our work are the city's local specialties, and that is of particular interest for travelers.

3. System Overview

This section first gives the annotations and formulates the local specialties mining problem, and then describes the overview of the algorithm.

3.1. Problem Formulation. Restaurant information often has similar structure in different local review applications. Figure 1 shows two typical local review applications: the left one is Yelp [1] and the right one is Baidu Shenbian [3]. Local mobile applications always associate business entities with locations, so the restaurant page is fixed to a city. A restaurant page has restaurant's name, user-generated rank, basic information, and the restaurant's featured dishes.

Each page in the local review website contains a ⟨city, restaurant, dishes⟩ hierarchical structure, so from restaurant pages of many cities, a hierarchical structure depicted in Figure 2 can be constructed.

The annotations of cities, restaurants, and dishes are listed in Table 1.

The local specialty mining problem is that: given a city c_i in C, rank the dishes in city $c_i\{d_1, d_2, \ldots, d_k, \ldots\}$, $d_k \in r_{ij}$ and $r_{ij} \in c_i$, so that dishes ranked higher are specialties in city c_i, which means these dishes are famous in city c_i, but seldom found in other cities.

3.2. System Architecture. Figure 3 illustrates the overview of the proposed method. Our local specialties mining and recommendation system consists of four steps: step 1, rank dish with structured local review data; step 2, rank dish with unstructured user-generated content; step 3, combine the ranking scores generated by steps 1 and 2; step 4, recommend local specialties and their associated restaurants, Q&A threads, and travelogues to a user. The local specialty mining algorithm will be elaborated in Section 4, and the recommendation service will be presented in Section 5.

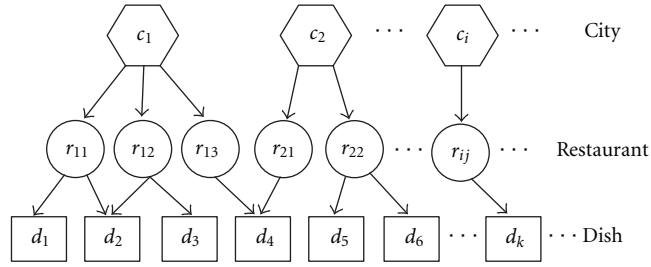

Figure 2: Hierarchical structure of local review data.

Table 1: Annotations.

Annotation	Meaning
$C = \{c_1, c_2, \ldots, c_N\}$	C: city collections. C_i: the ith city
$R_i = \{r_{i1}, r_{i2}, \ldots, r_{iM}\}$	R_i: restaurants in city C_i. r_{ij}: the jth restaurant in city C_i
$D = \{d_1, d_2, \ldots, d_k\}$	D: the dish collection. d_i: the ith dish name

4. Mining Local Specialties

This section elaborates the local specialty mining algorithm, which consists of 4 phases. (1) Filter dish to remove trivial dishes, noises and spams, and so forth. (2) Rank dish using local review data. (3) Calculate dish-city correlation using unstructured UGC. (4) Combine ranking scores generated by phases 2 and 3 to generate local specialties, and merge duplicates and rerank the top local specialties.

4.1. Dish Filtering. Since the dishes recommended in the restaurant page are often extracted from user's comments or added by restaurant owner, their quality cannot be assured. They must be filtered first during the dish mining process.

We consider a specialty's quality is low and must be discarded if it is in the two following cases.

(1) The specialty is trivial and not informative.

(2) The specialty is spam/noisy/meaningless.

We develop two simple rules to filter the specialty with low quality. For case 1, if the dish name often occurs in other dish name, then this dish name is not informative and can

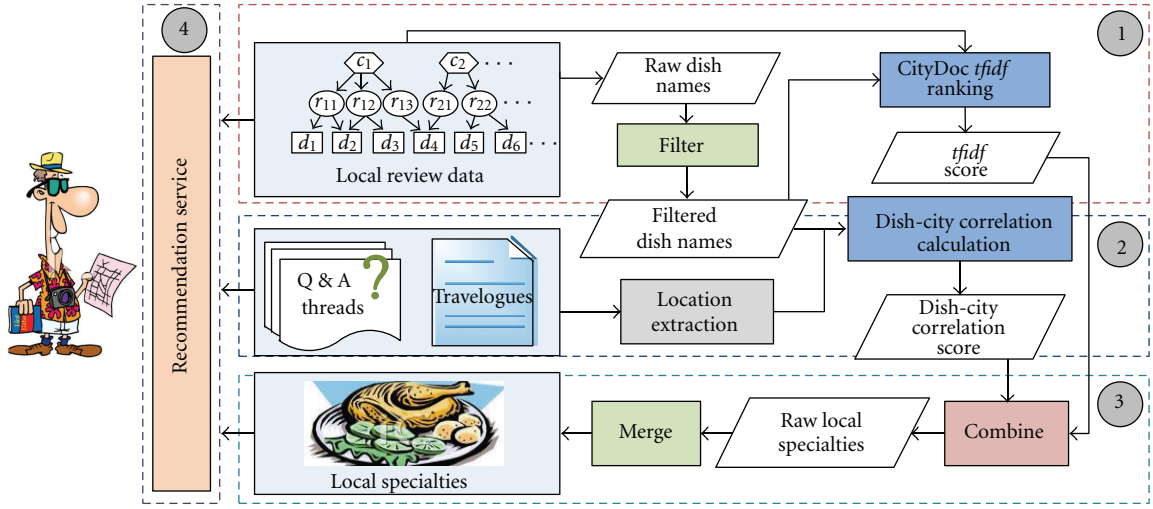

Figure 3: System overview.

be discarded. For example, *fried rice* often occurs in many other dish names, such as *Yangzhou fried rice*, *beef fried rice*, *egg fried rice*, and *shredded pork fried rice*, so the dish *fried rice* should be filtered out. For case 2, if the dish name is only recommended by one restaurant, then it should be discarded.

4.2. Dish Ranking Using Local Review Data. As what can be seen from Figure 2, a restaurant can only recommend a dish once, but a dish can be recommended by more than one restaurant, thus a dish may be recommended several times in a city. Intuitively, if a dish is recommended many times by different restaurants in one city, but seldom recommended in other cities, then this dish might be the city's local specialty. So the question becomes how to find dishes that frequently appear in one city but seldom appear in other cities? And, how to analytically measure a dish's "appear in one city with high frequency but appear in other cities with low frequency"? To answer this question, the *tfidf* weighting is naturally brought into use.

The *tfidf* weighting (term frequency-inverse document frequency) algorithm [15] is often applied in text analysis and document retrieval task to select representative words for a document. A word will have a high *tfidf* weight if it matches the following two conditions: (1) the word has a high term frequency (*tf*) in the given document; (2) the word has a low document frequency (*df*) in the whole collection of documents, thus has a high inverse document frequency (*idf*). This exactly fits the local specialty mining problem, if we consider the city as a document, and the dishes as words. The *tf* can measure how frequently a dish is recommended in a city, while the *idf* measures how seldom the dish is recommended in other cities. So we adopt the *tfidf* weighting to rank dishes as follows.

For each city c_i, we concatenate the dishes recommended by restaurants in this city as a *city document*:

$$CityDoc(c_i) = \text{concatenate}(d_k), \quad \text{where } d_k \in r_{ij},\ r_{ij} \in c_i. \tag{1}$$

If a dish is recommended several times by different restaurants in a city, then this dish will occur the same times in the city document. For example, city c_1 in Figure 2 has the city document in the following form:

$$CityDoc(c_1) = \{d_1, d_2, d_2, d_3, d_4\}. \tag{2}$$

After *CityDoc* is built for each city, *tfidf* weighting can be applied to rank dishes in the cities:

$$tfidf(c_i, d_k) = \frac{\#d_k}{|CityDoc(c_i)|} * \log\left(\frac{|C|}{\#c_j : d_k \in CityDoc(c_j)}\right). \tag{3}$$

In which $\# d_k$ stands for the times that dish d_k occurs in $CityDoc(c_i)$, $|CityDoc(c_i)|$ is the document length of $CityDoc(c_i)$, $|C|$ is the total city count, and $\#c_j$ is the amount of *CityDoc* that contains dish d_k.

After the *tfidf* ranking, a dish d with higher *tfidf (c,d)* is more likely to be local specialty in city c.

4.3. Leverage Unstructured UGC. Since cuisine hunting is often an important task for travelers, a traveler may wonder what is the local specialty in her travel destination when she is planning her trip. She may resort to community Q&A website for help, such as Yahoo! Answers [4] and Quora [16]. For example, she might ask "what dish I have to try when I travel to Beijing", and someone may answer "You should definitely try the *Beijing roast duck*". And after the trip, she might like to write travelogues to share her experience, including the local specialty she has enjoyed in the travel destination. In this way, the location "*Beijing*" and the dish name "*Beijing roast duck*" may cooccur many times in Q&A threads and travelogues.

So it is reasonable to exploit the information hidden in community Q&A websites and travelogues to help the local specialty mining task. A straightforward idea is to use

the location and dish cooccurrence to measure the correlation of a dish and a location.

Due to the nature of social network applications, the question and answers tend to be short in community Q&A websites, therefore it might be easy to extract location from a community Q&A thread. But to identify a travelogue's associated location is a much harder task. The length of a travelogue is often long, and the description in a travelogue often contains many details. For example, a traveler may mention her starting location of her trip, locations along the trip, and the destination of the trip; furthermore she may compare the destination to some locations she has traveled before. As a result, a travelogue may mention multiple locations. So after locations are extracted from a travelogue, it is necessary but difficult to identify which location is the travel destination and is emphasized in the travelogue. In this paper, we follow the work in [9] to identify the location emphasized in a travelogue. In [9] locations are first extracted from the travelogues, and two types of features are calculated for these locations, textual features and geographical features. After that, to leverage the two independent types of features, a cotraining framework is adopted to build a classifier to identify a travelogue's emphasized location.

After locations are identified in Q&A threads and travelogues, the correlation between a dish and a city can be measured as

$$\mathrm{corr}(c,d) = \frac{co(c,d)}{\sum_{c\in C} co(c,d)}. \quad (4)$$

In which $co(c,d)$ stands for the cooccurrence of the dish d and a city c in both Q&A website and travelogues, and C stands for the city collection.

The final ranking score for a dish to be a city's local specialty can be obtained by combining the *tfidf* weight from local review data and the correlation scores from user-generated content:

$$w(c,d) = \lambda * tfidf(c,d) + (1-\lambda) * \mathrm{corr}(c,d). \quad (5)$$

In which λ is a factor that controls the combination of structured local review data and unstructured UGC.

4.4. Merge Duplicate Dishes. Some dish may be referred to as alias or abbreviations in local review data's restaurants' featured dishes, Q&A threads, and travelogues. That might cause duplicate dishes in mined city specialties. For example, *Beijing roast duck* and *roast duck* refer to the same dish in Beijing, and they are all mined as Beijing's local specialties. It would be awkward and confusing if both of them are recommended to users, so we developed an algorithm to tackle this problem.

We observed that dish names often consist of 4 parts: ingredients, flavors, cooking methods, and other auxiliary words. We believe if the ingredients, flavor, and the cooking method are the same in different dish names, these dish names should refer to the same dish. Take the *Beijing roast duck* and *roast duck* as an example, the cooking method *roast* and the ingredient *duck* are the same in these two dish names, so these two dish names are considered to refer to the same dish. The auxiliary word *Beijing* is ignored in the dish name matching process.

We crawled dish ingredients, flavors, and cooking methods from several cooking recipe websites such as [17], and we used these words as to segment dish names. If the ingredients, flavor, and the cooking method are the same in different dish names, these dish names are considered to refer to the same dish and should be merged. The dish name with highest weight $w(c,d)$ is chosen as the exemplar of the merged dish names, and merged dish names' weights are accumulated as the exemplar's weight. For example, if Beijing's ranked local specialties {(dish, w (Beijing, dish))} are {(*noodles with soybean paste, 0.18*), (*Beijing roast duck, 0.15*) (*fried pork tripe, 0.12*), and (*roast duck, 0.08*)}, then *Beijing roast duck* and *roast duck* will be merged and *Beijing roast duck* will be chosen as exemplar and their weights are accumulated. So the final ranked local specialties list will be {(*Beijing roast duck, 0.23*) (*noodles with soybean paste, 0.18*), (*fried pork tripe, 0.12*)}.

5. Recommendation Service

After local specialties have been mined for each city, we build a recommendation service, so that a traveler can easily find the city's local specialty and the associated restaurants, Q&A threads, and travelogues. That brings great convenience to travelers for the cuisine hunting and decision making. The recommendation service's user interface is given in Figure 4. When a user selects a city, the city's local specialty will be listed (Figure 4(a)). For each local specialty, the user can browse associated restaurants (Figure 4(b)), related Q&A threads (Figure 4(d)) and travelogues (Figure 4(c)).

A local specialty's associated restaurants are restaurants whose featured dishes contain the local specialty. We adopted HITS algorithm [18] with small modification to rank these restaurants considering both user-generated rank and the restaurant-dish relations. In the ranking algorithm, the restaurants are considered as *hubs* and the dishes are considered as *authorities*, and the *hub* scores of restaurants are initialized with restaurants' user-generated rank. The tradeoff between user-generated rank and restaurant-dish relations is controlled by iteration steps of *hubs* and *authorities* calculation. Convergence of iteration leads to a ranking result fully focused on restaurant-dish relation, and early stop in the iteration step can get a ranking result biased to user-generated rank. After the ranking procedure, a restaurant with a high *hub* score tends to have more popular dishes and a higher user-generated rank. This should be more preferred for a traveler.

Snippets of Q&A threads and travelogues will be presented to the user after a local specialty is specified. The local specialty's related Q&A threads and travelogues are first ranked by term frequency of the dish, and then the surrounding texts are extracted to form the snippets. These snippets serve as a great compensation for the restaurants list from local review application data. They can help the travelers a lot, because they are written by travelers and experienced users and contains more vivid and detailed information such

Figure 4: Local specialty recommendation service. (a) City's local specialties. (b) Dish's-associated restaurants. (c) Dish-associated travelogues. (d) Dish associated Q&A threads.

as how the dish tastes, which restaurant services the best dish, and how to get there.

This recommendation service has been integrated into the MSRA travel guide project: http://travel.msra.cn/.

6. Experiments

This section first describes the settings of experiments, and compares the proposed method to other three methods. The combination factor of local review data and unstructured UGC is also evaluated. The results are reported in details, followed by some discussions.

6.1. Data. We crawled 380965 restaurant pages from Baidu Shenbian, and extracted the ⟨city, restaurants, dishes⟩ hierarchical structure from them. Furthermore, we crawled 182706 location related community Q&A threads from popular Q&A websites such as Zhidao [19], Wenwen [20] and iAsk [21], and 324905 travelogues from travelogue sharing websites and such as Sina travel blog [22], Netease travel blog [23], and Lvping [24]. For the dish duplicates merging in Section 4.4, we crawled a dataset containing 1723 ingredients, 68 cooking methods, and 50 flavors from cooking recipe websites.

Figure 5 depicts the top 15 cities that have the most restaurants. The amount of restaurants in a city reveals the city's geographic scale and its economic development.

Figures 6 and 7 give the top 15 cities according to Q&A thread count and travelogue count. These counts reveal the cities' popularity of tourism.

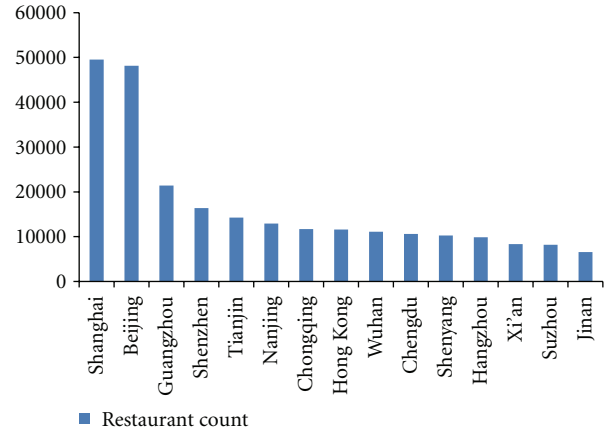

Figure 5: Top 15 cities according to restaurant count.

6.2. Comparison Methods. In order to investigate the effectiveness of mining local specialty by leveraging both structured local review data and the unstructured user-generated content, we employ other three methods for comparison purpose. So there are total four methods which are evaluated in the experiments.

(i) *LocalReivew_HITS*: This method utilizes the restaurant-dish relationship of local review data to rank restaurants and dishes in each city. It employs HITS algorithm [18] on the restaurant-dish graph by considering the restaurants as *hubs* and the dishes as

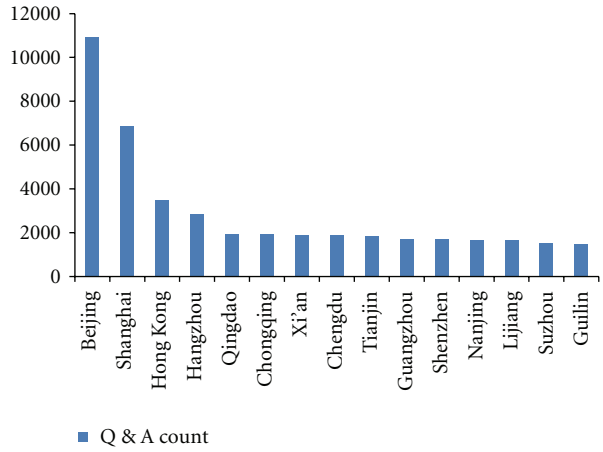

Figure 6: Top 15 cities according to Q&A count.

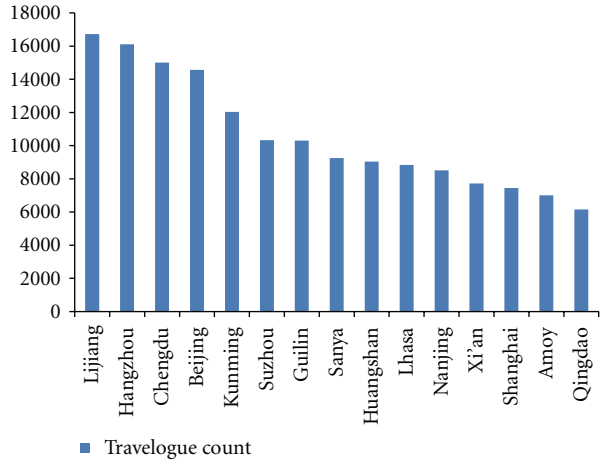

Figure 7: Top 15 cities according to travelogue count.

authorities, and the *hub* scores are initialized with restaurants user-generated rank. After the iterative score propagation is done, dishes with high *authority* scores are considered as the city's specialties. The basic assumption behind this algorithm is that a city's specialties are offered in large amount of restaurants in this city, especially the high-ranked ones, thus the specialties will have high *authority* scores. We refer to this method as *LocalReview_HITS*.

(ii) *LocalReview_tfidf*: This method groups all dishes provided by a city's restaurants as a document, and then *tfidf* weighting algorithm is applied on all city documents. Dishes with high *tfidf* weight are considered as a city's specialties. This method is elaborated in Section 4.2.

(iii) *Unstructured UGC*: In this method, the local review data is only used to build dish name dictionary to detect dish in unstructured user-generated content (UGC), that is, travelogues and Q&A threads. The correlation of dishes and cities are mined from unstructured UGC, and dishes having high correlation to a city are considered as the city's specialties. This method is discussed in Section 4.3.

(iv) *Combination*: This method is the local specialty mining algorithm proposed in this paper. It combines both the *tfidf* weight generated by *LocalReview_tfidf* and the city-dish correlation generated by *UnstructuredUGC*.

6.3. Combination Factor. The factor λ controls the tradeoff between the influences of two different data sources, that is, the structured local review data and unstructured UGC. We change the factor λ in a range from 0 to 1 with a step size of 0.1 and evaluate the local specialty mining performance with respect to λ. In this way, we can investigate how this factor affects the algorithm's performance and the contribution of two different data sources.

6.4. Evaluation. We select top 15 cities according to restaurant count, Q&A thread count, and travelogue count, respectively, and obtain a city set that contains 23 unique cities. They are Beijing, Shanghai, Hong Kong, Guangzhou, Shenzhen, Sanya, Chongqing, Lijiang, Hangzhou, Suzhou, Guilin, Nanjing, Xi'an, Shenyang, Jinan, Lhasa, Tianjin, Kunming, Huangshan, Chengdu, Qingdao, Wuhan, and Amoy. The evaluation of different local specialty mining algorithms is performed in these cities.

Since it is very difficult to find all the specialties in a city, even for human, the recall is hard to measure. So we only investigate the precision of mining algorithms here. Average precision [25], which is a widely used evaluation metric in information retrieval research community, is adopted to measure the effectiveness of different local specialty mining algorithms. The ground truth are manually labeled by people with domain knowledge.

6.5. Results and Discussions. Table 2 lists top 5 local specialties in some major cities which are generated by our proposed method. The Chinese dish names and their English translations are given, and the correct specialties are marked as italic.

Figure 8 shows the average precision of cities' top 5 dishes (AP@5) and top 10 dishes (AP@10) generated by different algorithms. λ is set as 0.4 in the *Combination* method. Table 3 shows the mean average precision over the 23 cities to demonstrate overall performance of different mining algorithms.

From Figure 8 and Table 3, we can tell that *LocalReview_HITS's* performance is the worst, and *LocalReview_tfidf* works much better. This tells us that the assumption behind *LocalReview_HITS* is wrong, that is, local specialties are not provided by large amounts of local restaurants, which means a traveler cannot easily find local specialties just by browsing top restaurants in a local review website. This is especially the case in large cities such as Beijing, Shanghai, Guangzhou and Hangzhou. Because in these cities, the geographical scale is large, the economy is highly developed, and the residents are of various culture backgrounds, and these factors lead to

Table 2: Examples of mined cities' specialties. The italic dish names are the correct ones.

City	Top 5 specialties				
Beijing	*Beijing roast duck* (北京烤鸭)	*Noodles with soybean paste* (炸酱面)	*Fried pork tripe* (爆肚)	*Stewed liver* (炒肝)	*Fermented bean drink* (豆汁)
Shanghai	*Friedbun* (生煎)	Tripe (肚子)	Hamburger (汉堡)	*Nanxiang steamed small bun* (南翔小笼)	Apple (苹果)
Hong Kong	*Eggette* (鸡蛋仔)	*Dessert* (甜品)	*Wonton noodle* (云吞面)	*Fish ball* (鱼蛋)	Water convolvulus (通菜)
Hangzhou	*Shrimp with longjing tea leaves* (龙井虾仁)	*SongSao fish soup* (宋嫂鱼羹)	*West lake fish in vinegar gravy* (西湖醋鱼)	*Dongpo pork* (东坡肉)	*Pianerchuan noodle* (片儿川)
Suzhou	*Whitebait* (银鱼)	Lvyang Wonton (绿杨馄饨)	*Zhuangyuan Pork Knuckle* (状元蹄)	*fermented bean curd* (臭豆腐)	*shoe-shaped crispy cake* (袜底酥)
Nanjing	*Soup with duck flood and vermicelli* (鸭血粉丝汤)	*Chicken gravy dumpling* (鸡汁汤包)	Lion bridge (狮子桥)	*Boiled salted duck* (盐水鸭)	*Shredded bean curd* (干丝)
Xi'an	*Pita bread soaked in lamb soup* (羊肉泡馍)	*Chinese hamburger* (肉夹馍)	*Fried rice with pickled vegetable* (酸菜炒米)	*Cold rice noodle* (凉皮)	Gravy dumpling (汤包)
Tianjin	*Goubuli stuffed bun* (狗不理包子)	*Thin pancake with puffed fritter* (煎饼果子)	*Twist of dough* (麻花)	*Fried cake* (炸糕)	*Shredded mung bean pancake* (锅巴菜)
Kunming	*Puer tea* (普洱茶)	*Bridge rice noodles* (过桥米线)	*Chicken cooked with potato* (洋芋鸡)	*Yunnan rice cake* (粑粑)	*Yunnan cheese* (乳扇)
Amoy	*Ip's glutinous rice cake with sesame* (叶氏麻糍)	*Sweet herb jelly* (烧仙草)	*Sateysauce noodle* (沙茶面)	*Oyster omelet* (海蛎煎)	*Peanut soup* (花生汤)

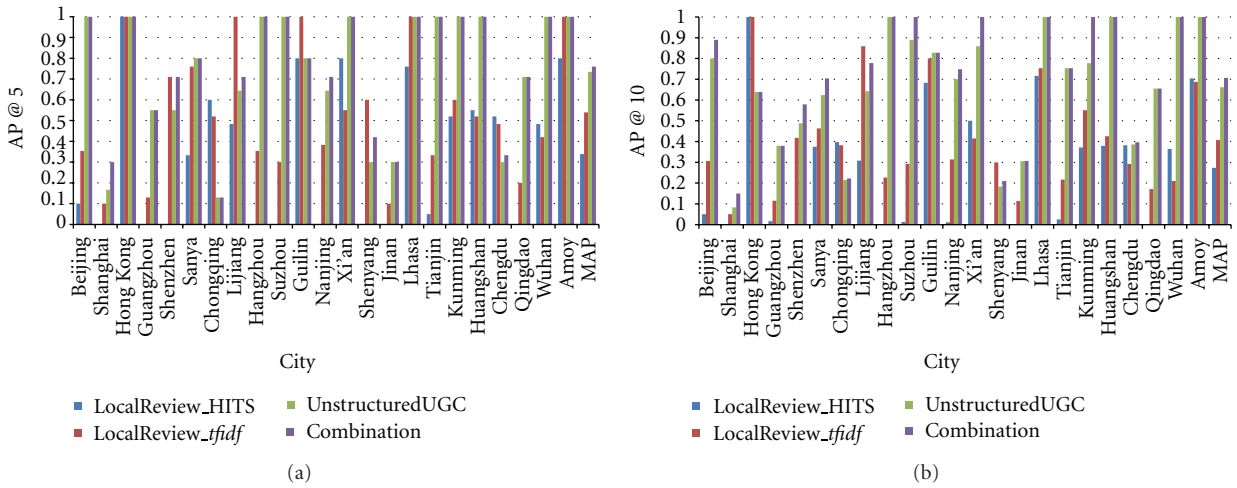

Figure 8: Average precision of top cities.

Table 3: Mean average precision of different algorithms.

MAP	*local reivew_hits*	*Local Review_*tfidf	Unstructured UGC	Combination
MAP@5	0.339	0.540	0.735	0.760
MAP@10	0.274	0.407	0.661	0.706

restaurants with enormous quantity and style. Algorithms leveraging unstructured user-generated content, that is, *UnstructuredUGC* and *Combination*, works much better than algorithm using only local review data. This result can be intuitively explained because the Q&A threads and travelogues reveal more information that is specifically related to travelers and tourism. The *Combination* method works best in all evaluated algorithms, which proves the effectiveness of our proposed algorithm.

Figure 9 shows the performance changing of the proposed method with respect to the combination factor λ. The figure tells that (1) mining from unstructured UGC alone (λ is 0) can achieve a better performance than mining from the structured local review data alone (λ is 1), (2) when the contribution of local review data increases (a larger λ), the performance gets better, but when the local review data's contribution increases to a certain threshold (λ grows above 0.7), the performance degrades dramatically. That means the structured local review data and the unstructured UGC can mutually reinforce the mining performance, and the unstructured UGC plays a more important role in the proposed algorithm. The local review data alone is insufficient,

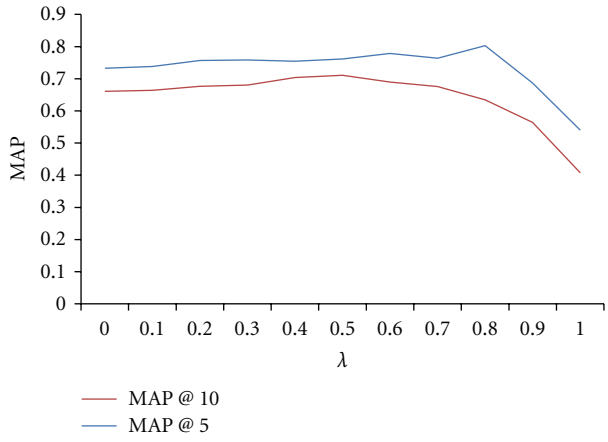

Figure 9: MAP w.r.t. combination factor λ.

but with the help of unstructured UGC, the performance can be boosted a lot.

7. Conclusion and Future Work

This paper proposes a mining algorithm to deal with the novel local specialty mining problem, which is of particular interest for travelers. The proposed algorithm leverages both structured data from local review websites and unstructured data from user-generated content from Q&A websites and travelogues. We first extract dish names from local review data to build a document for each city, and apply *tfidf* weighting algorithm on these documents to rank dishes. Dish-city correlations are calculated from unstructured UGC, and combined with the *tfidf* ranking score to discover local specialties, followed by duplicates removal. Finally a recommendation service is built to present local specialties to travelers, along with specialties'-associated restaurants, Q&A threads, and travelogues. Experiments on a large data set demonstrate the effectiveness of the proposed algorithm. The results show that, the proposed algorithm can achieve a good local specialty mining performance. And compared to using local review data alone, leveraging unstructured UGC can boost the mining performance a lot, especially in large cities.

In the future, we intend to continue this research in 2 directions. (1) Exploiting the hierarchical structure of the local review data more thoroughly, such as investigating the relation between restaurants and the relation between dishes. (2) To study the unstructured UGC in the semantic level to get a better insight of travel related information.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no. 60933013), National Science and Technology Major Project (Grant no. 2010ZX03004-003), and Fundamental Research Funds for the Central Universities (Grant no. WK2100230002). This work is performed at MSRA.

References

[1] Yelp, http://www.yelp.com/.
[2] Dianping, http://www.dianping.com/citylist.
[3] Baidu Shenbian, http://s.baidu.com/city.
[4] Yahoo! Answers, http://answers.yahoo.com/.
[5] Y. Gao, J. Tang, R. Hong, Q. Dai, T. S. Chua, and R. Jain, "W2Go: a travel guidance system by automatic landmark ranking," in *Proceedings of the 18th ACM International Conference on Multimedia ACM Multimedia*, pp. 123–132, October 2010.
[6] Yahoo! Travel, http://travel.yahoo.com/.
[7] R. Ji, X. Xie, H. Yao, and W. Y. Ma, "Mining city landmarks from blogs by graph modeling," in *Proceedings of the 17th ACM International Conference on Multimedia, MM'09, with Co-located Workshops and Symposiums*, pp. 105–114, October 2009.
[8] Windows Live Space, https://login.live.com/.
[9] M. Ye, R. Xiao, W. C. Lee, and X. Xie, "On theme location discovery for travelogue services," in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*, pp. 465–474, 2011.
[10] Q. Hao, R. Cai, J. M. Yang et al., "TravelScope: standing on the shoulders of dedicated travelers," in *Proceedings of the 17th ACM International Conference on Multimedia, MM'09, with Co-located Workshops and Symposiums*, pp. 1021–1022, October 2009.
[11] Q. Hao, R. Cai, C. Wang et al., "Equip tourists with knowledge mined from travelogues," in *Proceedings of the 19th International World Wide Web Conference (WWW '10)*, pp. 401–410, April 2010.
[12] Q. Hao, R. Cai, X. J. Wang, J. M. Yang, Y. Pang, and L. Zhang, "Generating location overviews with images and tags by mining user-generated travelogues," in *Proceedings of the 17th ACM International Conference on Multimedia, MM'09, with Co-located Workshops and Symposiums*, pp. 801–804, October 2009.
[13] T. C. Peng and C. C. Shih, "Mining Chinese restaurant reviews for cuisine name extraction: an application to cuisine guide service," in *Proceedings of the International Conference on Information Engineering and Computer Science (ICIECS '09)*, pp. 1–4, December 2009.
[14] C. C. Shih, T. C. Peng, and W. S. Lai, "Mining the blogosphere to generate local cuisine hotspots for mobile map service," in *Proceedings of the 4th International Conference on Digital Information Management (ICDIM '09)*, pp. 151–158, November 2009.
[15] G. Salton, E. A. Fox, and H. Wu, "Extended Boolean information retrieval," *Communications of the ACM*, vol. 26, no. 11, pp. 1022–1036, 1983.
[16] Quora, http://www.quora.com/.
[17] Douguo, http://www.douguo.com/.
[18] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.
[19] Baidu Zhidao, http://zhidao.baidu.com/.
[20] Soso Wenwen, http://wenwen.soso.com/.
[21] Sina iAsk, http://iask.sina.com.cn/.
[22] Sina travel blog, http://blog.sina.com.cn/lm/travel/.
[23] Netease travel blog, http://blog.163.com/travel.html.
[24] Lvping, http://www.lvping.com/.
[25] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, UK, 2008.