Abstract: Choosing a restaurant is one of the most frequent decisions faced in modern daily life; however, it is difficult for consumers to choose between food/restaurant by reading large amounts of reviews. This study attempts to generate cuisine hotspot maps through blog content mining to help consumers make restaurant decisions by specialties. The main obstacle in doing this involves recognizing and extracting restaurants and essential restaurant information (i.e., restaurant dishes) in unstructured content. In contrast to traditional Named Entity Recognition (NER) targets, dish name is a promising target that received little attention in previous studies. This study develops methods for recognizing and extracting restaurant names and dish names from review posts in the blogosphere and achieves satisfactory performance. Based on the method, we processed more than 12,000 Chinese blog posts and generated a cuisine hotspot map. The map shows the most popular dishes of restaurants in a map-view to help consumers make restaurant decisions. A prototype of cuisine hotspot map, named CuisineGuide, is implemented and available as an iPhone application.

Categories and Subject Descriptors
H.5.2 [User Interfaces]: Natural Language: 1.2.7 Natural Language Processing; [Text Analysis]

General Terms: Blogs, Chinese blog posts, iPhones, Geographical Information Systems

Keywords: Web search, blog analysis, Web content, Hotspots, Name entity recognition

Received: 11 May 2010; Revised 14 July 2010; Accepted 20 July 2010

1. Introduction

Individuals share opinions of stores or products in online posts, which collectively reflect the “ground-truth” opinions of Internet users. The ground-truth opinions, also known as “word of mouth,” are dominating consumer purchase decisions. A market research was conducted surveying 1,200 online consumers in 2008, and demonstrated that over 80% of online consumers decide between two or three products based on consumer reviews [5].

It is difficult for consumers to choose between products by reading large amounts of reviews. Consumer reviews should be processed in advance to give consumers a good comparison base. For example, suppose that people usually choose restaurants according to the most distinguished cuisine, cuisine name provided by restaurants should be summarized beforehand, enabling consumers to quickly find best choices.

This study attempts to generate cuisine hotspot maps through blog content mining. The main obstacle in doing this involves extracting and recognizing restaurants and essential restaurant information (i.e., restaurant dishes) in unstructured content. In contrast to traditional Named Entity Recognition (NER) targets, dish name is a promising target that received little attention in previous studies. This study develops methods for recognizing and extracting restaurant names and dish names from Chinese blog posts. The results are arranged into hotspots (hot restaurants and dishes) and presented in map views.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 describes the system architecture. Section 4 discusses the implementation details and evaluation results. Section 5 describes a prototype of cuisine hotspot map. Conclusions are finally drawn in Section 6, along with future directions.

2. Related work

Tezuka et al. [9] argued that current integration of web search with GIS only links local search results to the map, which limits the capability for the user in many cases. For a tighter integration of web content with GIS, the proposed three levels of prototype systems, such as extraction, knowledge discovery, and presentation, were implemented and evaluated in their study. The first prototype system in the extraction level aimed to extract the experience of people at sightseeing spots from blog posts. A blog map of experiences was generated by adopting association rule mining and some heuristic rules to find hot objects under the given time and place. Their work was similar to our study. However, since they could only extract hot objects on a local region from blog content, the type of objects (e.g., schools, restaurants, movie theaters and museums) is unable to be distinguished for different applications. In contrast, by using a blog post classifier and the proposed domain-specific methods, we can not only extract hotspot restaurants precisely, but also provide hot cuisine information for each hot restaurant respectively. In addition, their evaluation result only showed a very low precision (0.216) for the extraction task, which was in practice less useful than our work.

Currently, many commercial websites focus on collecting restaurant reviews. These websites collect reviews by encouraging users to share reviews (e.g., Yelp1) or aggregating existing online users review articles (e.g., Google Maps) or doing both (e.g., Yahoo! Local). Map interfaces are provided in all mentioned websites. Users can look up restaurants by locations, categories, ratings, and other features. However, none of them can list the popular dishes of restaurants.

3. System implementation

This section proposes a system workflow for identifying cuisine hotspots (that is, popular restaurants and popular dishes) from blog posts. The workflow is divided into three parts, as follows:

1http://www.yelp.com
1) Filtering unqualified posts: Although the blogosphere is a rich source of review posts, the discrepancy of post quality would cause a “garbage-in-garbage-out” problem. We consider three dimensions to filter out unqualified posts. First, because hotspots vary over time and we are finding the latest hotspots, outdated posts are considered as unqualified posts. Second, posts that contain very few words are considered as invalid posts. Third, because a post referred to a restaurant name is not definitely talking about cuisine of the restaurant, mechanisms are required to filter out unrelated posts.

2) Recognizing restaurants and dishes from blog posts: In this part, two tasks are executed – restaurant name recognition and dish name recognition. In restaurant name recognition task, given a list of restaurant names, we detect if a post mentions any known restaurant names. Because it is very often that bloggers mention more than one restaurant in their blog posts, even though the main topic target of a post is only focused on parts of the mentioned restaurants, we develop heuristics for solving the problem. In dish name recognition task, we try to find out dish names in a post based on a dish name corpus. Because it is not possible to exhaustively collect dish names, we characterize known dish names and use the characteristics to find unknown dish names.

3) Discovering cuisine hotspots: In this part, blog posts are transformed into triplets (Post, Restaurant, Dish) and are aggregated to discover hotspots. Two kinds of cuisine hotspots are discovered: hot restaurants and hot dishes in a restaurant.

Figure 1 illustrates the workflow. Implementation details are discussed in the subsequent sections.

3.1 Cuisine-related Blog Post Classifier

3.1.1 Blog dataset crawling and filtering

We randomly crawled 5,500 blog posts from wretch.cc, which is the biggest blog service provider in Taiwan. Blog posts that contained fewer than 50 words or were published more than two years previously were discarded. The resulting dataset comprised 3,281 posts. For the training task, 1,602 blog posts were assigned to two categories, cuisine and non-cuisine, respectively. The rest posts were used as testing data. No overlap exists between the training and testing data. Note that the gathered blog posts are not only used for training cuisine-related blog post classifier. The extracted cuisine-related blog posts can be assumed to have targeting restaurant names and dish entities inside text; therefore, these posts can further be labeled and used for evaluating the performance of our proposed restaurant name detection and dish entity recognition algorithms in later sections.

3.1.2 Chinese word segmentation and feature selection

The lack of delimitation of words makes Chinese word segmentation more difficult than that in other languages (namely, English). However, the accuracy of Chinese word segmentation influences the performance of Chinese information retrieval significantly [11]. This study adopted a conditional random field segmentor proposed by Chang et al. [10]. Additionally, we also utilized a PAT-tree-based keyword extraction tool introduced by Chien [2], which makes use of significant lexical pattern extraction for word segmentation, to extract frequent words. The resulting number of segmented words for the blog post dataset is 596,532 words.

To find representative features for classification, this study adopted chi-square (CHI) statistics as the ranking metric, which is shown to be one of the most effective feature selection methods for text classification [12]. However, the chi-square metric is known to be unreliable for rare terms [7]. Therefore, based on the segmented words ranked by chi-square values, we considered only words with DF >= 5 as the representative features for classification.

3.1.3 Evaluating classification model

Support Vector Machine (SVM) was applied as the classifier for blog post classification [7, 13]. For the learning process of SVM classifier, this study used LIBSVM\(^{2}\) [1], a famous software for SVM classification, to efficiently train the SVM model. We adopted the following TF/IDF formula as weighting scheme for each feature term \(w_i\):

\[
w_i = (1 + \log n_i) \log \frac{N}{n_i}
\]

where \(n\) denotes the number of documents and \(n_i\) represents the number of documents in which a feature term \(i\) occurs at least once. Furthermore, \(n_i\) is the term frequency of term \(i\) in document \(d\), and \((1 + \log n_i)\) denotes a variation of TF used for smoothing. The evaluation result demonstrated that the SVM classification model performed well by achieved an optimal \(F_1\) score of 0.870. By using the SVM classifier, 1,679 posts were processed and classified for later tasks.

3.2 Restaurant name detection

3.2.1 Restaurant data source

This study used a Point of Interests (POI) database established by Kingwaytek Technology\(^{3}\) as the target restaurant data source. The database contains ordinary and geographical information on approximately 9,819 restaurants in Taiwan, such as restaurant name (in Chinese or English), sub-branch address, telephone, website and corresponding geographic position. Only blog posts mentioning restaurant names included in POI database were processed.

3.2.2 Mapping blog posts to restaurants

Given a list of restaurants in a POI database, the simplest method of detecting restaurant names in blog posts is through absolute string matching of post content. However, this method

\(^{2}\)http://nlp.stanford.edu/software/segmenter.shtml

\(^{3}\)http://www.csie.ntu.edu.tw/~cjlin/libsvm/

\(^{4}\)Official website: http://www.kingwaytek.com/
fails to detect the correct number of target restaurants in some cases that multiple topics are mentioned simultaneously in post content.

To solve this problem, this study proposed a heuristic method for measuring the confidence of string matching results. First of all, after manually examining more than 2000 cuisine-related Chinese blog posts, we observed that target restaurant names are usually mentioned in the following sections of blog posts:

- **title**: it indicates the main topic of the post.
- **first paragraph**: the main topic target often stated at the beginning of the post.
- **last paragraph**: after sharing the experiences about restaurants, bloggers sometimes would append some contact information for readers, such as the name, address, telephone number and opening hours of the commented restaurants.

If a matched restaurant name appears in one of the above sections in the blog post, it should be more likely to be the target. Therefore, to follow this rule, each matched restaurant $s_i$ is assigned a positional score value $\text{PosScore}(s_i)$ according to its absolute index position in the blog post. Any matched restaurant name that appears in one of the above sections will be assigned a score of 1, or otherwise a score of 0.5.

$$\text{PosScore}(s_i) = \begin{cases} 1 & \text{if } pos(s_i) \in P_{\text{title-first-last}} \\ 0.5 & \text{otherwise} \end{cases}$$

However, since there might still be some cases that do not follow the proposed rule, it is necessary to measure the confidence for detected results as well. A confidence scoring function is defined as:

$$\text{Confidence}(m_j) = \sum_{i=1}^{n} \text{PosScore}(s_i), m \equiv B \rightarrow R$$

where $m_j$ represents a mapping pair $B \rightarrow R$ from a blog post $B$ to a restaurant $R$, and $n$ denotes the occurrence of a restaurant name in a given post. $\text{Confidence}(m_j)$ is computed by the sum score of $\text{PosScore}(s_i)$.

### 3.2.3 Evaluating the performance

In the blog post dataset, which is introduced in Section 3.1.1, 723 post-restaurant pairs were manually labeled to evaluate the performance of the proposed method. The baseline for comparison is the direct string matching method that simply matches restaurant names from the full text of a post. As shown in Table 1, the proposed mapping method, which combined a heuristic rule and occurrence scoring methods, outperformed the baseline. Among all evaluated methods, the combination approach yielded a 70.3% improvement in precision when compared with the baseline, and achieved a best $F_1$ score of 0.802.

### 3.3 Dish entity recognition

Most NER research studies in English and Chinese focused on recognizing names of persons, organizations and locations, and numeric entities including time, date, and so on [3]. In contrast to the entities mentioned above, the dish name entity is an important and interesting entity that was neglected in early research studies. Recently, a study by Rennie and Jaakkola [6] paid attention to the task of recognizing restaurant name entity and pointed out the high application value of NER in the restaurant domain, for instance, providing more timely feedback and information about restaurants from daily communication of Internet users compared to publishers of newspaper and restaurant reviews (e.g., Zagat.com).

In fact, dish name entities usually comprise meaningful elements that give us featured cues for this recognition task:

- **Ingredients** the main ingredients of a dish.
- **Culinary manner** how a dish is cooked or seasoned.
- **Cooking equipment** the type of equipment used for cooking a dish.
- **Origin** whether a dish originated from.
- **Appearance** description of the appearance of a dish.
- **Taste** description of the taste of a dish.
- **Transliteration** transliterated dish elements.

For example, **法式芝士火鍋** (French Cheese Fondue) is a French dish name comprising ingredients (**芝士** cheese), cooking equipment (**火鍋** Fondue) and origin (**法式** French). That is, dish names are basically compound names that comprise cuisine-related elements. Based on the above observations, this study aims to extract representative “dish features” reflecting dish elements with the aid of statistical techniques and dish corpus, and further to combine these features into a compound dish name. Figure 2 shows the proposed procedure for dish entity recognition.

### 3.3.1 Dish name corpus

Before starting dish entity recognition, a dish name corpus is required for extracting representative dish features. This study
3.3.2 Extracting dish features
For identifying "frequent n-gram patterns" that appear regularly in dish names, Mutual Information (MI) is suitable for this task, and has been widely used in Chinese word segmentation since the study of Sproat and Shih (1990) [8, 11]. MI measures the strength of association between two adjacent Chinese characters. The following formula is used:

\[ MI(c_i, c_j) = \log_2 \left( \frac{Nfreq(c_i, c_j)}{freq(c_i) \times freq(c_j)} \right) \]

where \( c_i \) and \( c_j \) are Chinese characters, \( freq(c_i, c_j) \) denotes the frequency of occurrence of adjacent characters \( c_i \) and \( c_j \), \( freq(c_i) \) and \( freq(c_j) \) represent the frequency of occurrence of characters \( c_i \) and \( c_j \), respectively, and \( N \) is the size of corpus.

If the MI score of a bigram \( c_i, c_j \) is below a predefined threshold \( T_{mi} \), the index point between characters \( c_i \) and \( c_j \) would be a segmentation point. For example, The dish 鮮食噴魚肚 (Stewed Fish Maw with Shredded Chicken) can be segmented into the following bigrams: 鮮食 (MI=1.701), 食噴 (MI=0.84) and 魚肚 (MI=1.463). Thus, if \( T_{mi} = 1.0 \), the dish is segmented into three sub-words [鮮食][噴魚][魚肚] representing different dish elements. This technique yielded 6,271 features representing different dish elements in the dish name corpus.

3.3.3 Serialized n-gram grouping & weighting
Given a collection of N-gram features, a compound dish name \( T_j \) can be created by grouping a set of adjacent N-gram features \( g_i \), which can be represented as:

\[ T_j = (g_1, g_2, g_3, ..., g_n) \]

Notably, overlap may exist between grouped N-gram features. This study defined a score function which considers the importance of each \( g_i \) and the prefix and suffix character of the compound. The weight of each \( g_i \) is calculated by Gain [4]. \( D \) denotes the number of dishes in corpus and \( d_i \) represents the document (dish name) frequency of each N-gram term \( g_i \):

\[ w(g_i) = \text{Gain}(g_i) = \frac{d_i}{D} \left( \frac{d_i}{D} - 1 - \log \frac{d_i}{D} \right) \]

Prefix/suffix weighting is based on the observation that the prefix and suffix character of Chinese dishes are often representative dish features, for example, the suffix character (rice) in 香菇牛肉炒飯 and the prefix character (許) in 許家天婦羅. Therefore, this study defined \( 1/ \log(N/d_{\text{prefix}}) \) and \( 1/ \log(N/d_{\text{suffix}}) \) as the prefix and suffix weights for each generated compound \( T_j \), where \( N \) denotes the size of dish name corpus, while \( d_{\text{prefix}} \) and \( d_{\text{suffix}} \) represent the occurrence of the prefix and suffix of \( T_j \) respectively. The detail of the score function \( \text{Score}(T_j) \) is denoted as follows:

\[ \text{Score}(T_j) = \alpha \sum_{i=1}^{n} w(g_i)^{1/k} + (1-\alpha) \left( \frac{1}{\log(N/d_{\text{prefix}})} + \frac{1}{\log(N/d_{\text{suffix}})} \right) \]

where \( \alpha \) denotes a coefficient for regulating the mixture ratio between two weighting methods, and \( k \) stands for the length of the N-gram feature \( g_i \). Furthermore, \( 1/k \) is used to balance the influence of short and long N-grams on \( T_j \). All compounds with scores exceeding a threshold \( T_{core} \) are extracted.

Since the number of extracted dish features is restricted by the size of the dish name corpus, if a given blog post contains some dishes composed of unknown dish features, those dishes may not be extracted correctly. For example, consider the following case:

**Chinese:** 我點了一道青椒牛肉炒飯，試試它的口味。

**English:** I ordered a dish named "green pepper and beef fried rice", and wanted to try out its taste. the dish 青椒牛肉炒飯 (Green Pepper and Beef Fried Rice), which consists of three features, such as 青椒 (pepper), 牛肉 (beef) and 炒飯 (fried rice), may be wrongly extracted as 牛肉炒飯 if the feature 青椒 is not included in the dish feature set.

However, based on the language specificity of the Chinese, the unknown feature problem can be alleviated by utilizing the technique of Chinese word segmentation into the process of n-gram feature grouping and weighting. That is, any segmented compound words containing at least one known dish feature are also put into the scoring function. Therefore, the score of segmented compound words is calculated by their prefix/suffix characters and known features. This idea is similar to the technique of training dataset extension used in traditional data mining. In the previous case, if the sentence is segmented correctly (as underlined words), the dish can be extracted by treating the segmented compound 青椒牛肉炒飯 as a candidate dish compound grouped by known dish features, even though the feature word 青椒 is unknown in the gathered dish corpus. Then, we can compute the \( \text{Score}(\text{青椒牛肉炒飯}) \) to determine whether to extract the candidate dish.

The effect of utilizing segmented words into the process of dish entity recognition is evaluated in the next section.

3.3.4 Evaluating the performance
For evaluation of the proposed method, this study manually labeled 1,271 dishes from 100 blog posts in our blog dataset (The blog dataset is introduced in Section 3.1.1). The compared methods are as follows:

- **Corpus Only:** extracting dish names by direct dish name matching. No weighting mechanism used.
- **Corpus-based Feature Grouping:** extracting dish names by grouping nearest corpus feature terms.
- **Gain-weighted (G-W) Feature Grouping:** using only a Gain-weighted score for recognizing possible dish names.
- **Prefix/suffix-weighted (P/S-W) Feature Grouping:** using only a prefix/suffix-weighted score for recognizing possible dish names.
- **G-W+P/S-W Feature Grouping:** combining Gain weighting with prefix/suffix weighting mechanism.
- **Last three methods with segmented words used:** utilizing segmented words for dish feature grouping.

The performance of the proposed methods was assessed in terms of precision, recall, F, and exclusion rate. The exclusion rate measures the percentage of non-NEs that are correctly filtered. The performance results are summarized in Table 2. The evaluation results revealed that the proposed method, which combines Gain-weighting and prefix/suffix weighting mechanism, works well for the task. Interestingly, as shown in Figure 3, correctly segmented Chinese words are very useful for boosting the performance of dish entity recognition. Note that
segmented words are not always correct, since they are often segmented by rule-based or statistical algorithms. However, utilizing such segmented words can expand the machine readable scope of n-gram dish features appeared discontinuously in the text. This combined approach is adapted in the prototype system demonstrated in Section 4, which takes only about 0.6 second for processing a blog post.

<table>
<thead>
<tr>
<th>Compared Methods</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
<th>Exclusion rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Corpus Only (direct dish name matching)</td>
<td>0.508</td>
<td>0.399</td>
<td>0.447</td>
<td>0.963</td>
</tr>
<tr>
<td>Corpus-based</td>
<td>0.302</td>
<td>0.790</td>
<td>0.437</td>
<td>0.832</td>
</tr>
<tr>
<td>G-W ($T_{score} = 0.8$)</td>
<td>0.583</td>
<td>0.755</td>
<td>0.658</td>
<td>0.950</td>
</tr>
<tr>
<td>P/S-W ($T_{score} = 0.8$)</td>
<td>0.435</td>
<td>0.774</td>
<td>0.557</td>
<td>0.907</td>
</tr>
<tr>
<td>G-W+P/S-W ($T_{score} = 0.8, \alpha=0.3$)</td>
<td>0.871</td>
<td>0.501</td>
<td>0.636</td>
<td>0.984</td>
</tr>
<tr>
<td>G-W (with seg. words , $T_{score} = 0.8$)</td>
<td>0.452</td>
<td>0.906</td>
<td>0.603</td>
<td>0.900</td>
</tr>
<tr>
<td>P/S-W (with seg. words , $T_{score} = 0.8$)</td>
<td>0.290</td>
<td>0.914</td>
<td>0.440</td>
<td>0.801</td>
</tr>
<tr>
<td>G-W+P/S-W (with seg. words, $T_{score} = 0.8, \alpha=0.3$)</td>
<td>0.758</td>
<td>0.851</td>
<td>0.802</td>
<td>0.974</td>
</tr>
</tbody>
</table>

Table 2. Comparison of the performance for dish entity recognition

![Figure 3. P-R curve w/wo segmented words](image)

**3.4 Hotspot discovery**

Through the processes presented in Section 4.1 - 4.3, blog posts are transformed into Post-Restaurant-Dish triplets, which provide the basis for hotspot discovery. This study discovers two kinds of cuisine hotspots, that is, hot restaurants and hot dishes of a restaurant.

Given the following notations: a set of Post-Restaurant-Dish triplets $T$ contains triplets $\{ t_1, t_2, \ldots, t_n \}$, $freq(t)$ is the frequency that $t$ occurs in $T$, $R(t)$ is the referring restaurant in triplet $t$, and $D(t)$ is the referring dish in triplet $t$. This study defines the hotness function of a restaurant $r$, as the following:

$$Hotness(r) = \sum_{R(t)=r} freq(t)$$

Furthermore, the hotness function of a dish $d$ in restaurant $r$, is defined as the following:

$$Hotness(d, r) = \sum_{R(t)=r, D(t)=d} freq(t)$$

Simply, this study calculates numbers of distinct posts for each restaurant or restaurant-dish combination to derive the hotness index.

**4. Prototype system**

Figure 4 shows the screenshots of CuisineGuide, a prototype of mobile hotspot maps we implemented as an IPhone application. The data of CuisineGuide comes from 12,000 blog posts, which are collected from Xuite\(^3\). Hundreds of hot restaurants are chosen, and each of hot restaurant is given a mined hottest dish of the restaurant.

![Figure 4. Screenshots of CuisineGuide](image)

Figure 5 shows the usage flow of CuisineGuide. When users open the application, hot restaurants around the user are shown on the map. A restaurant list, as well as mined hottest dishes of each listed restaurant, is shown on the left side of the map to help user choose restaurants by dishes (Step 1). If users are interested in the provided essence information, detailed information, including contact information of the restaurant and top-N hottest dishes, will be shown after a click (Step 2). If users need further support, review articles are available for users as well (Step 3).

![Figure 5. Usage flow of CuisineGuide](image)

**5. Conclusion and future directions**

This study employs a scenario to explain why current mobile map services fail to present information that consumers can use to make purchase decisions. “Word of mouth,” which dominates online consumer purchase decisions, needs to be extracted and organized to be presented in mobile map services. This study thus tries to identify hotspots from blog content and integrate hotspot information into mobile map services. This study proposes and implements a system for extracting and

\(^3\)http://blog.xuite.net

400 Journal of Digital Information Management  Volume 8 Number 6  December 2010
inferring cuisine hotspots from Chinese blog posts and achieves satisfactory performance. The proposed feature elements for dishes also provide meaningful cues regarding the task of dish entity recognition in other languages.

In the future, we plan to extract more dynamic POI information from blog content, including popular time slots (such as morning, afternoon, evening and night) and opinions about restaurants to provide more fine-grained hotspot queries. Extracted information can also be fed back to POI databases, and thus a brand-new POI database comprising information extracted from UGC can be realized.

7. Acknowledgements

This study is conducted under the "Digital Convergence Service Open Platform" project of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

References