Big Social Data Special Issue

JIS

Journal of Information Science
2015, Vol. 41(6) 830–854

DOI: 10.1177/0165551515603323
jis.sagepub.com
SAGE

Where to go and what to play: Towards summarizing popular information from massive tourism blogs

Hualin Xu
University of Electronic Science and Technology of China, China

Hua Yuan
University of Electronic Science and Technology of China, China

Baojun Ma
Beijing University of Posts and Telecommunications, China

Yu Qian
University of Electronic Science and Technology of China, China

Abstract

In this work, we propose a novel method to summarize popular information from massive tourism blog data. First, we crawl blog contents and segment them into semantic word vectors separately. Then, we select the geographical terms in each word vector into a corresponding geographical term vector and present a new method to explore hot tourism locations and, in particular, their frequent sequential relations from a set of geographical term vectors. Third, we propose a novel word vector subdividing method to collect local features for each hot location, and introduce the metric of max-confidence to identify the Things of Interest (ToI) associated with the location from the collected data. We illustrate the benefits of this approach by applying it to a Chinese online tourism blog dataset. The experimental results show that the proposed method can be used to explore hot locations, as well as their sequential relations and corresponding ToI, efficiently.

Keywords

Blog mining; information retrieval; max-confidence; things of interest; travel sequence

1. Introduction

In recent years, tourism has been ranked as the foremost industry in terms of volume of online transactions [1] and most tourism sites (such as blog.tripadvisor.com and www.travelblog.org) enable consumers to post blogs to exchange information, opinions and recommendations about the tourism destinations, products and services within web-based communities. Here is an example from a tourism blog[1]:

Example 1. *Fly to **Mykonos** which has been selected as the top destination for luxury life-style with flocking in from around the world. **Little Venice** is a colourful neighbourhood in **Mykonos Town** with wooden balconies hanging above the sea. There you will find countless bars and café from which you can enjoy the most spectacular sunset.*

Meanwhile, some readers are more likely to enjoy a high quality travel experience from others' blogs. By obtaining reference knowledge from these blogs, individuals are able to visualize and manage their own travel plans. For instance,

Corresponding author:
Hua Yuan, School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 611731, China.
Email: yuanhua@uestc.edu.cn

a person is able to find some places that attract him from other people's travel routes, and schedule an efficient and convenient (even economic) path to reach these places. As a prospective tourist, people may want to know mainly two types of information from the online blogs:

- How did the others plan their wonderful trips (location selecting and route planning) for a targeted travel destination?
- What is particularly fun (Things of Interest [ToI]) in a selected location, for example, a scenic spot or a special restaurant?

The first type of information is about how to schedule an efficient travel route with a series of locations. It would help people in making better decisions about 'Where to go' in the target destination. Whereas, the second is about the detailed information of ToI associated with the selected locations and it would help people to make decisions about 'What to play' at each location.

However, from the perspective of the recommender role of tourism blogs, the blog readers may be confronted with an information overload when a large number of blog data are offered. In other words, it is impossible for the average readers to find out the exact common information they are interested in from such a huge mass of data. To solve this problem, data mining technology, such as documentation classification (or document categorization), has been introduced as a powerful tool into the task of blog information extracting in literature [2, 3]. The basis for such a task is about assigning a document to one or more classes or categories basing on some selected features. In the classification algorithm, the documents are usually represented by a vector space model [4]. The model has two limitations: the first is about the order in which the terms appear in the document is lost in the vector space representation [5, 6], and the second is it theoretically assumes terms are statistically independent [7], however, not all the words in a blog are necessarily independent variables [8]. Therefore, these two deficiencies would result in a lot of noise when we conduct the document classification algorithm into the work of blog information extraction directly.

In this work, by deeming each tourism location in one's targeted destination as a travel 'topic', and the ToI as some special interested local features associated with the location, we propose a research framework to summarize the popular tourism information from blogs as a whole. The contributions of this paper to the information retrieval from massive blogs lie in three aspects as follows:

- First, all the contents recorded by a blogger have their intrinsic sequenced relationship, for the tourism blogs, such a sequenced relationship could be in accordance with the tour route that the blogger had experienced. Therefore, mining frequent sequenced patterns from blog contents may reveal more information. In this work, we present a simple but efficient method to mine the frequent patterns of travel sequences from the massive blog contents.
- Second, a tourism blog/document typically concerns multiple topics. So, it is a challenging task to identify the local features for each topic from a set of blog generated word vectors. In this work, we propose a novel vector subdividing technology to construct a subsets of local features for each hot location according to the context of blog writing, from which we can identify the ToI associated to the specific hot location exactly. The significant result of this method is shielding the impacts of irrelevant word co-occurrences which have very high frequency.
- Third, we present a new method basing on the measurement of max-confidence to identify the popular ToI for each hot location from its local features. Moreover, we provide a rational lower bound of the range of max-confidence in text information summarizing.

For **Example 1**, if we find two hot locations of 'Mykonos' and 'Little Venice' from massive blog contents, then 'Mykonos – Little Venice' could be a travel route while they are mentioned in blogs frequently and sequentially. Also, {'Top destination', 'Life-style', 'Flocking', 'World'} and {'Neighbourhood', 'Wooden balconies', 'Sea', 'Bars', 'café', 'Sunset'} are the local features associated to these two locations, respectively. Further, if we can identify the popular local feature set (ToI) for 'Mykonos' and 'Little Venice' as {'Top destination', 'Flocking'} and {'Sea', 'Bars', 'café', 'Sunset'} respectively, then we could summarize them into Figure 1, which could provide the more concise but focused popular tourism information about this area.

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 sketches out the research framework. Section 4 discusses how to crawl raw blog contents online. Section 5 presents the method of hot locations and travel routes mining in tourism blogs. Section 6 discusses the ToI extraction method in detail. Section 7 shows the experimental results. This paper is concluded in Section 8.

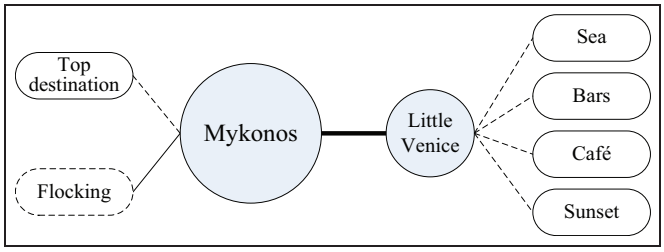


Figure 1. Locations, travel route and ToI in Example 1.

2. Related work

In this section, we briefly review the important studies related to this research.

2.1. Blog summarization and text mining

In [9], the authors presented a method to extract the temporal discussions, or stories, occurring within blogger communities, based on some query keywords. Paper [10] presented the opinion search engine for the TREC 2006 Blog Task which is based on an enhanced index on words denoting. Paper [3] proposed a shallow summarization method for blogs as a pre-processing step for blog mining which benefits from specific characteristics of the blogs.

In the blog summarization literature, we note that the basic technology used in online text processing is text mining [11, 12], which is used to derive insights from user-generated contents and primarily originated in the computer science literature [2, 13]. Thus some previous text mining research was focused on automatically extracting the opinions of online contents [14, 15] and the hot topics [16]. These methods used in blog mining not only involve reducing a larger corpus of multiple documents into a short paragraph conveying the meaning of the text, but also focus on features or objects on which customers have opinions. In particular, some research has been focused on mining tourism blogs for better successors' decision-making [17, 18].

2.2. Topic model and feature selection

A topic model is a type of statistical model for discovering the 'topics' that occur in a collection of documents and topic modelling is a way of identifying patterns in a corpus. An early topic model was probabilistic latent semantic indexing (PLSI), created by Thomas Hofmann [19] and the Latent Dirichlet Allocation (LDA), perhaps the most common topic model currently in use, is a generalization of PLSI developed by David Blei et al. in 2003 [20]. LDA is an unsupervised learning model basing on the intuition that documents are represented as mixtures over latent topics where topics are associated with a distribution over the words of the vocabulary. Therefore, it is good at finding word-level topics [21] and there were many proposed latent variable models based on it [22].

Another method is to look through a corpus for the clusters of words and group them together by a process of similarity or relevance, for example, frequent pattern analysis [23] and co-expression analysis [24, 25]. In these methods, a text or document is always represented as a bag of words which raises two severe problems: the high dimensionality of the word space and the inherent data sparsity. In literature, feature selection is an important technology used to deal with the problems [26].

Feature selection is a process that chooses a subset from the original feature set according to some criterions. The selected 'good' features should retain the original physical meaning and provide a better understanding for the data. There are lots of feature selection metrics assess the representativeness or importance of different document features, some of them are summarized in Table 1.

Obviously, all these methods are involved in feature selection for document classification. However, there are few impressive researches on providing blog readers valuable knowledge that they are personally interested in, i.e. the common topics from massive contents, as well as the special local features of these topics rather than that of the entire document.

3. The methodology

The research problem is to find: (1) a set of hot tourism locations as well as popular travel sequences among these locations; and (2) a set of ToI for each location. The following is about the research methodology.

Table 1. Main metrics for feature selection.

ID	Feature selection metric	Denotation
1	Term frequency and inverse document frequency	TF-IDF [5, 27]
2	Document frequency	DF [28]
3	Information gain	IG [29, 30]
4	Mutual information	MI [31, 32]
5	Chi-square	χ^2[31, 33]
6	Correlation coefficient	CC [34]
7	Odds ratio	OR [29]

3.1. Problem statement

Given a set of tourism blogs $\mathbb{B}$, and assume that the i-th blog in it can be represented by a *word vector*, i.e. $\boldsymbol{b}_i$, thus $\mathbb{B}$ can be represented with a set of vectors as $\boldsymbol{B} = \{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_i, \ldots, \boldsymbol{b}_{|\mathbb{B}|}\}$, and the total items in $\boldsymbol{B}$ is Σ_B. There are two metrics:

- $supp(X)$(with a predefined threshold *mini_supp*), which is used to measure the frequency of itemset X; and
- $\theta_{\{x,y\}}$ (with a predefined threshold θ_0), which is used to measure the dependency of item y on x (or vice versa).

Then, the research problem can be specified as two subtasks:

- for the dataset $\boldsymbol{B}$, find out a set of *frequent geographical terms* $\{b^H\} \in \Sigma_B$ (i.e. *hot locations* mentioned in $\mathbb{B}$) such that $supp(b^H) \geq mini\_supp$. The position relations of these *frequent geographical terms* are studied as well;
- for each *frequent geographical term* b^H, find out an appropriate set of terms $\{b_i\} \in \Sigma_B$ such that $\theta_{\{b^H b_i\}} \geq \theta_0$.

3.2. Research framework

The presented research framework in this work is about three parts: *blog extraction and word segmentation* (BEWS), *frequent travel routes mining* (FTRM) as well as *interesting things detection* (ITD). The whole process is shown in Figure 2, in which the grey legend represents the data source or the calculation results, and the white part refers to the corresponding data processing.

The BEWS subsystem provides the basic data processing for the research. In BEWS, the raw data of blogs in a tourism website are crawled into a blog dataset first. Then, each piece of blog is segmented into a set of semantic words so that it can be transformed into a word vector, in which, the elements are only semantic words and necessary punctuation marks after data cleaning.

The FTRM subsystem is introduced to mine the travel route from the blog generated word vectors. First, the non-geographical terms are eliminated from each word vector with the help of a geographical name table (*GNT*) to form a geographical dataset. Then, the frequent pattern mining method is implemented to obtain the frequent geographical terms. Finally, correlation analysis is conducted on the dataset of these frequent geographical terms to obtain the travel routes.

The ITD subsystem is used to mining ToI for each hot location. First, a vector subdividing method is proposed to establish a dataset of cut vector for collecting the local features of each frequent geographical term. Then the max-confidence metric is introduced to identify the interesting local features as ToI for each hot location.

4. Blog contents extraction

In BEWS subsystem, three subtasks of blog extraction, word segmentation and data cleaning are involved to transform a piece blog into a *word vector*.

4.1. From blog to word vector

Web crawling technology can help people extracting information from the website. In this work, it used to obtain large-scale users generated blogs from a tourism website. All the blogs are crawled into an initial dataset of $\mathbb{B}$.

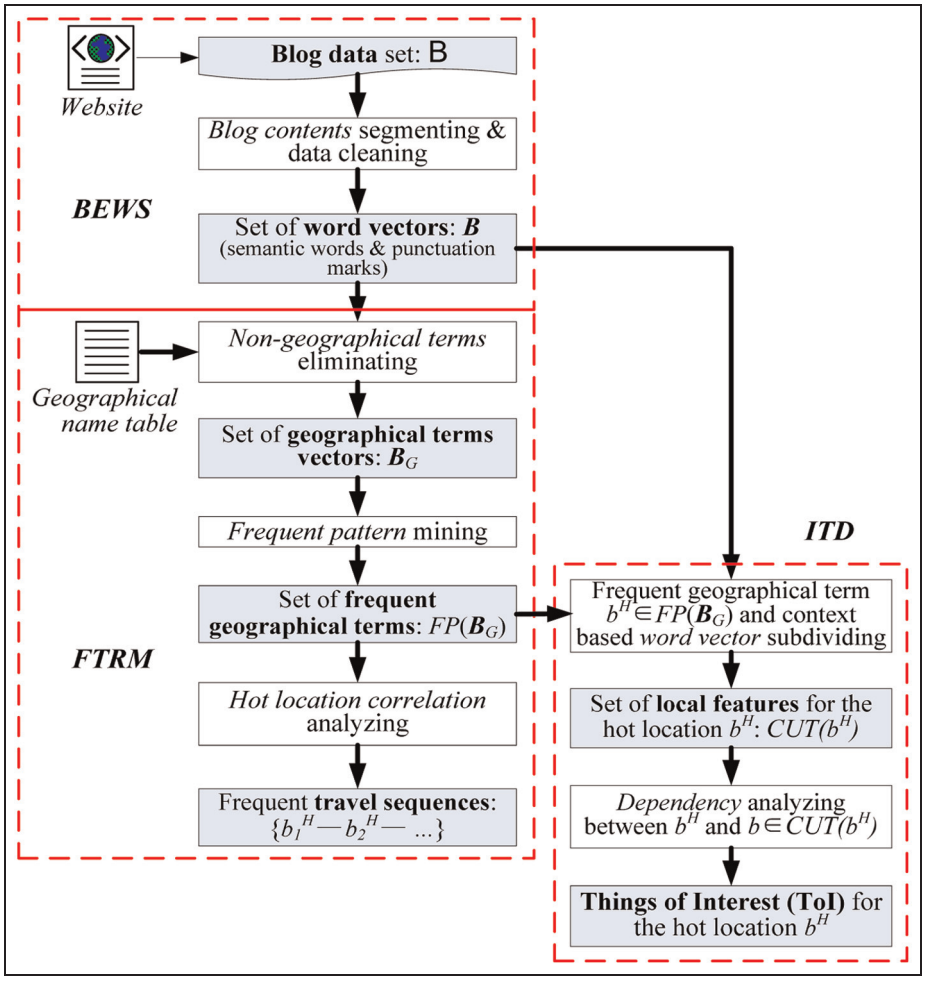


Figure 2. The method framework.

Word segmentation usually involves the tokenization of the input text into words at the initial stage of text analysis for NLP task [35]. In this work, it is the problem of dividing a string of written language into some component units.

For the work of data cleaning, we put it into an equivalent task on judging the usefulness of a component unit generated by the segmentation process. In general, the noise phrases, stop words and meaningless symbols are removed [36]. Here, we simply keep the following as the useful word segments:

- Semantic word (phrase) [36, 37]: Our goal is to find the hot tourism locations and the ToI associated with them. In tourism blogs, almost all of these two things are presented in the form of nouns.
- Punctuation marks [38]: In any text-based document, a period, a question mark or an exclamation mark is a real sentence ending. Therefore, people could take them as the sentence boundaries. We use symbol '||' to represent all types of these reserved punctuation marks.

Algorithm 1 shows the presented method as a whole. First, for each piece of blog in $\mathbb{B}$, its text contents are split into *word segments* based on some user-defined rules (Line 4). Then, the useless elements are cleaned and the remaining *word segments* are used to compose a vector $\boldsymbol{b}_i$ (Line 5–10). Finally, we obtain a set of *word vectors* as:

$$\boldsymbol{B} = \{\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{|\mathbb{B}|}\} = \bigcup_{i=1}^{|\mathbb{B}|} \{\boldsymbol{b}_i\}. \quad (1)$$

Algorithm 1. Blog extraction and pre-processing algorithm.

```
Input: Blog dataset 𝔹;
Output: Word vector B;
for i = 1 to |𝔹| do
    Split blog i in 𝔹 into n_i word segments {b_ij}, j = 1, ..., n_i;
    b_i = φ;
    for j = 1 to n_i do
        if b_ij is useful then b_i ← {b_ij}^2;
        end if
    end for
    B ← b_i;
end for
return B
```

The set of all items in $\boldsymbol{B}$ is $\Sigma_B = \bigcup_{i=1}^{|B|} \{b_{ij}\}$, where b_{ij} is the j-th term in $\boldsymbol{b}_i$. For **Example 1**, we can obtain a *word vector* of $\boldsymbol{b}$={*Mykonos, top destination, life-style, flocking, world || Little Venice, neighbourhood, Mykonos Town, wooden balconies, sea, bars, café, sunset*}.

4.2. Modified set operations for position-information-holding

In this work, we use 'vector' instead of the term of 'itemset' in the general data mining task, one main reason being to keep the positional relationship (relative position order) between the elements in $\boldsymbol{b}_i$.

Given two elements of $b_{is} \in \boldsymbol{b}_i$ and $b_{it} \in \boldsymbol{b}_i$ in the i-th blog, if the appearance of b_{it} is later than that of b_{is}, then the position relation of these element is $Pos(b_{is}) < Pos(b_{it})$. Here, $Pos(b_{ij})$ means to obtain the position information of element b_{ij} in vector $\boldsymbol{b}_i$.

To hold the elements position information in the blog generated *word vectors*, we enforce that the set operations in this work should be position-information-holding. That is, given a vector set of $X = \{x_1, \ \ldots, \ x_n\}$ and any set Y:

- Minus: $\boldsymbol{X} \backslash \{x_i\}_{x_i \in X} = \{x_1, \ \ldots, \ x_{i-1}, \ x_{i+1}, \ \ldots \ x_n\}$.
- Intersection: $\boldsymbol{X} \cap \boldsymbol{Y} = \boldsymbol{X} \backslash \cup \{x\}_{x \in X, x \notin Y}$.

5. Mining of frequent travel routes

In this part, we use the frequent pattern mining method to detect the hot tourist locations, i.e. *frequent geographical terms*, and propose a correlation analysing method to explore the travel routes between these hot locations.

5.1. Eliminating non-geographical terms

To find out the *frequent geographical terms* from the *word vectors*, a *geographical name table* is needed. This table can be extracted temporarily from an official travel guide providing by the local government, or provided by a creditable third part, for example, www.geonames.usgs.gov and maps.google.com. In particular, some tourism service websites like www.tripadvisor.com and www.ctrip.com can also provide a relative complete set of geographical terms for city tourism attractions (i.e. geographical term). It is noted that more than one source may be used in geographical terms extraction.

Given a *geographical name table* denoted by *GNT*, first we use it to filter out the non-geographical terms from $\boldsymbol{b}_i$ to form a *geographical term vector* as:

$$\boldsymbol{b}_{G_i} = \boldsymbol{b}_i \cap GNT \tag{2}$$

then we obtain a geographical dataset of $\boldsymbol{B}_G$:

$$\boldsymbol{B}_G = \bigcup_{i=1}^{|B|} \{\boldsymbol{b}_{G_i}\}. \tag{3}$$

Table 2. Dataset *B*.

Vector	Elements
$\boldsymbol{b}_1$	{ a1 A a2 || B b1 b2 || C c1 || D }
$\boldsymbol{b}_2$	{ A a1 a2 || B b1 || D || E }
$\boldsymbol{b}_3$	{ A a1 || b1 B b2 || C c1 A c2 }
$\boldsymbol{b}_4$	{ B b1 || D d1 E }
$\boldsymbol{b}_5$	{ a1 a2 A a3 || B b1 b2 || C c1 || F }

Table 3. Dataset B_G.

Vector	Elements
$\boldsymbol{b}_{G1}$	{ A B C D}
$\boldsymbol{b}_{G2}$	{ A B D E }
$\boldsymbol{b}_{G3}$	{ A B C A }
$\boldsymbol{B}_{G4}$	{ B D E }
$\boldsymbol{B}_{G5}$	{ A B C F }

Algorithm 2. Hot tourism location mining.

```
1: Input: Word vector set B, Geographical name table GNT, mini_supp;
2: Output: Hot tourism location set;
3: for i = 1 to |B| do
4:    b_Gi = b_i ∩ GNT;
5:    B_G ← b_Gi;
6: end for
7: Compute Σ_BG;
8: return FP^(1)(B_G).
```

Different from the traditional method, in $\boldsymbol{B}_G$ all the elements in the i-th record are the geographical terms that are mentioned in the i-th blog, and more importantly, the elements in each record should keep their positional order as in the original blog. All the items in $\boldsymbol{B}_G$ are $\Sigma_{B_G} = \bigcup_{i=1}^{|B_G|} \{b_{ij}\}$, where b_{ij} is the j-th term in $\boldsymbol{b}_{G_i}$.

Example 2. Given a $GNT = \{A,\ B,\ C,\ D,\ E,\ F\}$, and a dataset $\boldsymbol{B}$ composed of five word vectors (Table 2). We can obtain a geographical dataset with relation (2) as shown in Table 3.

5.2. Mining of hot tourism locations

From the perspective of *dataset* in database technology, *word vector* $\boldsymbol{b}_i \in \boldsymbol{B}$ is also a transactional data record, so is the *geographical term vector* of $\boldsymbol{b}_{G_i} \in \boldsymbol{B}_G$. Therefore, we can mine the itemsets that appear in $\boldsymbol{B}_G$ frequently as hot tourism locations for common people.

We denote the frequent n-itemset X in dataset $\boldsymbol{B}$ as follows:

$$FP^{(n)}(B) = \{X | X \subseteq \Sigma_B,\ |X| = n,\ supp(X) = \frac{\sigma(X|\boldsymbol{B})}{|\boldsymbol{B}|} \geq mini\_supp\}, \tag{4}$$

where $\sigma(X|\boldsymbol{B}) = |\{\boldsymbol{b}_i | X \subseteq \boldsymbol{b}_i, \boldsymbol{b}_i \in \boldsymbol{B}\}|$ means the *support count* [39] of X in dataset $\boldsymbol{B}$ and *mini_supp* is a predefined threshold.

Similarly, we can define $FP^{(n)}$ on any transaction dataset. In particular, the frequent 1-itemsets in $\boldsymbol{B}_G$, i.e. $FP^{(1)}(\boldsymbol{B}_G)$ can be seen as the hot tourism locations. In **Example 2**, we can obtain $FP^{(1)}(B_G) = \{A, B, C, D\}$ with $mini\_supp = 60\%$.

Algorithm 2 shows the detailed mining processes. Note that the frequent pattern mining technology is not the main concern in this work, thus any feasible algorithms can be used depending on the contents and data formation of $\boldsymbol{B}$.

Table 4. Dataset $\boldsymbol{B}_{G^H}$.

Vector	Elements
$\boldsymbol{b}_{G_1^H}$	{ A B C D }
$\boldsymbol{b}_{G_2^H}$	{ A B D }
$\boldsymbol{b}_{G_3^H}$	{ A B C A }
$\boldsymbol{b}_{G_4^H}$	{ B D }
$\boldsymbol{b}_{G_5^H}$	{ A B C }

5.3. Travel route generation

From a semantic perspective, the position relationship of all the elements in vector $\boldsymbol{b}_{G_i}$ shows the real travel sequence (location correlations) of blogger i. Thus, the common relations of all these geographical terms should lie in $\boldsymbol{B}_G$. Generally, the basic correlation between two locations is the co-occurrence and the adjacent position relationship of their representative terms in $\boldsymbol{B}_G$. Given two locations of $b_{is} \in \boldsymbol{b}_{G_i}$ and $b_{it} \in \boldsymbol{b}_{G_i}$, the adjacent position relationship means that b_{is} and b_{it} are mentioned frequently in $\mathbb{B}$ and $|Pos(b_{is}) - Pos(b_{it})| = 1$. In the following, we will propose a new method to reveal these location correlations.

First, we filter out all the unpopular locations (infrequent geography terms) from $\boldsymbol{b}_{G_i}$ to get a simplified *frequent geographical term vector*:

$$\boldsymbol{b}_{G_i^H} = \boldsymbol{b}_{G_i} \cap FP^{(1)}(\boldsymbol{B}_G). \tag{5}$$

Here, the elements in $FP^{(1)}(\boldsymbol{B}_G)$ indicate the common and frequent concerns of the bloggers (travellers). Thus, relation (5) tells us the *hot locations* mentioned in the i-th blog.

Lemma 1. $\boldsymbol{b}_{G_i^H} \subseteq \boldsymbol{b}_{G_i} \subseteq \boldsymbol{b}_i$.

Proof. According to relations (2) and (5), **Lemma 1** holds true.

Further, we can calculate:

$$\boldsymbol{B}_{G^H} = \bigcup_{i=1}^{|B|} \{\boldsymbol{b}_{G_i^H}\}. \tag{6}$$

All the items in $\boldsymbol{B}_{G^H}$ is $\Sigma_{\boldsymbol{B}_{G^H}}$.

Next, assume that m_i *hot locations* appear sequentially in the i-th blog:

$$\boldsymbol{b}_{G_i^H} = \{b_{i1}^H \; b_{i2}^H \; \ldots \; b_{im_i}^H\}. \tag{7}$$

To keep the position information of these *hot locations* in $\boldsymbol{b}_{G_i^H}$, we transform the formation of $\boldsymbol{b}_{G_i^H}$ into

$$\tilde{\boldsymbol{b}}_{G_i^H} = \{b_{i1}^H b_{i2}^H, \; b_{i2}^H b_{i3}^H, \; \ldots \; b_{i(j-1)}^H b_{ij}^H, \; b_{ij}^H b_{i(j+1)}^H, \; \ldots, \; b_{i(m_i-1)}^H b_{im_i}^H\}, \tag{8}$$

where "$b_{ij}^H b_{i(j+1)}^H$" in $\tilde{\boldsymbol{b}}_{G_i^H}$ means that two *hot locations* of b_{ij}^H and $b_{i(j+1)}^H$ are mentioned sequentially in blog $\boldsymbol{b}_i$. Similarly,

$$\tilde{\boldsymbol{B}}_{G^H} = \bigcup_{i=1}^{|\boldsymbol{B}|} \{\tilde{\boldsymbol{b}}_{G_i^H}\}. \tag{9}$$

For **Example 2**, if $FP^{(1)}(\boldsymbol{B}_G) = \{A, B, C, D\}$, then the calculation results of $\boldsymbol{B}_{G^H}$ and $\tilde{\boldsymbol{B}}_{G^H}$ are shown in Tables 4 and 5, respectively. In Table 5, if we set $mini\_supp = 40\%$, then $FP^{(1)}(\tilde{\boldsymbol{B}}_{G^H}) = \{AB, BC, BD\}$. Any element in $FP^{(1)}(\tilde{\boldsymbol{B}}_{G^H})$ indicates a correlation between two *hot locations* which are mentioned sequentially and frequently by bloggers.

From a network perspective, the *hot locations* in $FP^{(1)}(\boldsymbol{B}_G)$ and their correlations in $FP^{(1)}(\tilde{\boldsymbol{B}}_{G^H})$ can form a *route network* of $G = (V, E)$ by setting the vertex set as

$$V = FP^{(1)}(\boldsymbol{B}_G), \tag{10}$$

Table 5. Dataset B_G.

Vector	Elements
$\widetilde{\boldsymbol{b}}_{G_1^H}$	{ AB BC CD }
$\widetilde{\boldsymbol{b}}_{G_2^H}$	{ AB BD }
$\widetilde{\boldsymbol{b}}_{G_3^H}$	{ AB BC CA }
$\widetilde{\boldsymbol{b}}_{G_4^H}$	{ BD }
$\widetilde{\boldsymbol{b}}_{G_5^H}$	{ AB BC }

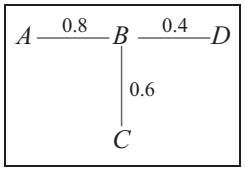


Figure 3. The weighted route network in Example 2.

and setting the edge set as

$$E = FP^{(1)}(\tilde{\boldsymbol{B}}_{G^H}). \tag{11}$$

In order to facilitate the calculation, the weight of edge $(b_{ij}, b_{i(j+1)})$ can be set as the *support* of item '$b_{ij}^H b_{i(j+1)}^H$' in $FP^{(1)}(\tilde{\boldsymbol{B}}_{G^H})$, i.e. $w_{(b_{ij}^H, b_{i(j+1)}^H)} = supp(\{b_{ij}^H, b_{i(j+1)}^H\})$. For the $FP^{(1)}(\tilde{\boldsymbol{B}}_{G^H})$ generated by the data in Table 5 ($mini\_supp = 40\%$), a weighted *route network* is formed illustrated in Figure 3, where *A, B, C* and *D* are hot locations.

Note that if the *support* of any two adjacent items in relation (8) are bigger than *mini_supp* simultaneously, the transformation process can be repeated to find the travel routes having more than 2 hops.

6. ToI extraction

Since a ToI is an interesting thing for a corresponding location, then it should be mentioned relatively frequently in context so that the location is the central topic in a set of blogs. Based on this idea, in this section we proposed a new method to extract ToI for each hot tourist location from the dataset $\boldsymbol{B}$.

6.1. Hot-location-based blog word vector subdividing

It is well known that if there are too many hot tourism locations and popular interesting things in the same *word vector*, the impact of noise correlations between elements is inevitable when mining frequent patterns directly from $\boldsymbol{B}$. Fortunately, bloggers tend to present something interesting around (or nearby) the appearances of a *hot location* in blogs. Moreover, if the contents of some adjacent sentences are about the same *hot location* (topic), then these sentences may constitute a topic domain for the location. Also, the topic domain is usually surrounded by a set of punctuation marks. We can expect that the closest local features (ToI) for each *hot location* would exist in such a topic domain and are bound by a pair of adjacent punctuation marks.

According to Lemma 1, the elements in $\boldsymbol{b}_i$ can be classified into two types (overlap is allowed): *hot locations* (i.e. elements also existing in $\boldsymbol{b}_{G_i^H}$) and the others. Therefore, $\boldsymbol{b}_i$ can be subdivided into sub-itemsets according to the topic boundaries of its m_i *hot locations* (see relation (7)). For the j-th hot location b_{ij}^H, we just need to cut the adjacent elements which belong to the same topic domain of b_{ij}^H. Since blog sentences usually end with a set of punctuation marks, the divided sub-itemset for b_{ij}^H can begin with the first punctuation mark before b_{ij}^H (the first term denoted by b_{ij_S}) and end with the first punctuation mark just before the next *hot location* of $b_{i(j+1)}^H$ (the last term denoted by b_{ij_E}):

$$\boldsymbol{b}_i = \{\ \ldots,\ \ b^H_{i(j-1)},\ \ \ldots, b_{i(j-1)_E} \Big\| \underbrace{b_{ij_S},\ \ldots, b^H_{ij},\ \ldots, b_{ij_E}}_{j-th\ sub-itemset} \Big\| b_{i(j+1)_S},\ \ldots,\ \ b^H_{i(j+1)},\ \ \ldots\}, \tag{12}$$

where $b^H_{ij} \in FP^{(1)}(\boldsymbol{B}_G), j = 1, \ldots, m_i$ and symbol '||' denotes the punctuation marks. According to the usual text writing style of bloggers, there is a high possibility that the ToI for location b^H_{ij} may lie in the itemset of $\{b_{ij_S}, \ldots, b^H_{ij}, \ldots, b_{ij_E}\}$.

More generally, given a *hot location* $b^H \in FP^{(1)}(\boldsymbol{B}_G)$, we define its *cut vector* in $\boldsymbol{b}_i$ as follows:

$$CUT(b^H|\boldsymbol{b}_i) = \{\ b_{ij_S},\ \ldots,\ b^H,\ \ \ldots,\ b_{ij_E}\}, \tag{13}$$

such that:

- $b^H \in FP^{(1)}(\boldsymbol{B}_G) \bigcap CUT(b^H|\boldsymbol{b}_i)$.
- $b_{i(Pos(b_{ij_S})-1)} = b_{i(Pos(b_{ij_E})+1)} =$ '||', which means the elements standing before b_{ij_S} and next to b_{ij_E} should be '||'.
- $Pos(b_{ij_S}) \leq Pos(b^H) \leq Pos(b_{ij_E})$.

The formed dataset of *cut vectors* for b^H from $\boldsymbol{B}$ is:

$$CUT(b^H) = \bigcup_i \{CUT(b^H|\boldsymbol{b}_i)\}. \tag{14}$$

All the items in $CUT(b^H)$ are $\Sigma_{CUT(b^H)} = \bigcup_{i=1}^{|CUT(b^H)|} \{b_{ij}\}$, where b_{ij} is the j-th term in $CUT(b^H|\boldsymbol{b}_i)$.

Obviously, there are two key elements in a *cut vector*, i.e. a *hot location* term as the semantic keyword, and two punctuation marks as the semantic topic boundaries for such a keyword. In fact, $CUT(b^H)$ can be deemed as a set of *local contexts* for term b^H which refers to either an ordered sequence or unordered set of other useful words in the same sentence (or in a set of adjacent sentences), such that they co-occur, have syntactic dependencies, or both [40, 41]. Consequently, we can expect that all the potential ToI about location b^H may lie in $CUT(b^H)$.

In **Example 2**, we know that *hot locations* are $FP^{(1)}(\boldsymbol{B}_G) = \{A, B, C, D\}$ while $mini\_supp = 60\%$, then we obtain the datasets of local features for all the *hot locations* (Table 6). A more general example is from the **Example 1**. If $FP^{(1)}(\boldsymbol{B}_G)$={Mykonos, Little Venice}, then the local features for 'Mykonos' are {*Mykonos, top destination, life-style, flocking, world*} and those for 'Little Venice' are {*Little Venice, neighbourhood, Mykonos Town, wooden balconies, sea, bars, café, sunset*}.

Note that there may be more than one *cut vector* generated for hot term b^H in the same blog $\boldsymbol{b}_i$, i.e. $|CUT(b^H|\boldsymbol{b}_i)| > 1$. In order to overcome the deficiency of data sparsity in $CUT(b^H)$, we merge these cut vectors for the same *hot location* into one. For example, if the j-th sub-itemset and k-th sub-itemset are two different *cut vectors* for the same hot term of b^H in $\boldsymbol{b}_i$:

$$\boldsymbol{b}_i = \{\ldots \| \underbrace{b_{ij_S},\ \ldots,\ b^H,\ \ \ldots,\ b_{ij_E}}_{j-th\ sub-itemset} \| \ldots \| \underbrace{b_{ik_S},\ \ \ldots,\ b^H,\ \ \ldots,\ b_{ik_E}}_{k-th\ sub-itemset} \| \ldots\}, \tag{15}$$

then, they will be merged into $CUT(b^H|\boldsymbol{b}_i) = \{\ b_{ij_S},\ \ldots,\ b^H,\ \ \ldots,\ b_{ij_E}, b_{ik_S},\ \ldots, b_{ik_E}\}$ (See $CUT(A|\boldsymbol{b}_3)$ in Table 6). This is also an efficient mechanism for preventing the overdone personal preferences (a blogger mentioned the same thing repeatedly) to affect the mining results. Accordingly, we obtain the following **Lemma 2**.

Table 6. Cut vectors for all the hot locations in Example 2.

Vector $\boldsymbol{b}_i$	$CUT(A\|\boldsymbol{b}_i)$	$CUT(B\|\boldsymbol{b}_i)$	$CUT(C\|\boldsymbol{b}_i)$	$CUT(D\|\boldsymbol{b}_i)$
$\boldsymbol{b}_1$	{a l A a2}	{B b l b2}	{C c l}	{D}
$\boldsymbol{b}_2$	{A a l a2 }	{B b l }	{}	{D }
$\boldsymbol{b}_3$	{A a l C c l A c2}	{b l B b2 }	{C c l A c2}	{}
$\boldsymbol{b}_4$	{}	{B b l}	{}	{D d l E}
$\boldsymbol{b}_5$	{a l a2 A a3}	{B b l b2}	{C c l}	{}

Lemma 2. $\sigma(b^H|\boldsymbol{B}) = |CUT(b^H)|$.

Proof. Knowing forms the definition of *support count*, $\sigma(b^H|\boldsymbol{B}) = |\boldsymbol{b}_i|b^H \in \boldsymbol{b}_i, \boldsymbol{b}_i \in \boldsymbol{B}|$. If $b^H \in \boldsymbol{b}_i$ and $\boldsymbol{b}_i \in B$, then we can get a (merged) $CUT(b^H|\boldsymbol{b}_i)$ according to relation (15). So, **Lemma 2** holds true.

Given any term $b \in \Sigma_{CUT(b^H)}$, the *cut vectors* for the 2-itemset $\{b^H b\}$ is defined as $CUT(b^H b) = \{CUT(b^H|b_i)|\boldsymbol{b} \in CUT(b^H|\boldsymbol{b}_i), CUT(b^H|\boldsymbol{b}_i) \in CUT(b^H)\}$, then we have the result of **Lemma 3**.

Lemma 3. $\sigma(\{b^H b\}|B) = \sigma(b|CUT(b^H))$.

Proof. According to **Lemma 2**, $\sigma(\{b^H b\}|B) = |CUT(b^H b)|$.

Further, $\sigma(b|CUT(b^H)) = |CUT(b^H|\boldsymbol{b}_i)|b \in CUT(b^H|\boldsymbol{b}_i), CUT(b^H|\boldsymbol{b}_i) \in CUT(b^H)| = |CUT(b^H b)|$. It holds true.

6.2. Dependency between two frequent co-occurring terms

6.2.1. Max-confidence. In real-world applications, many transaction datasets have inherently skewed support distributions which often lead to so-called 'cross-support patterns' [42]. The 'cross-support patterns' typically represent spurious associations among items with substantially different support levels.

Since we know that the support of words in blogs are distributed askew because some aspects are common (popular) topics for bloggers while others are not [43], then these 'cross-support patterns' may reflect a master-slave relationship between itemsets which can be used to reveal some semantic information in online information retrieval. For example, when bloggers record a tourist city (b_1), some of them tend to review a scenic spot (b_2) in b_1. In such cases, the appearances of the scenic spot ($supp(\{b_2\})$) are totally dependent on the blogs in which both the city and the scenic spot ($supp(\{b_1, b_2\})$) were reviewed. That is to say, the scenic spot has a strong relation (dependency) with the city, implying that if somebody inquires about city information, then the scenic spot as a necessary ToI of the city should be provided.

Here, we identify the dependent relation between b_1 and b_2 in the 2-itemset $X = \{b_1, b_2\}$ with a metric of *max-confidence*, which is defined as [44]:

$$\theta_X = \frac{supp(X)}{min\{supp(\{b_1\}), supp(\{b_2\})\}} = max\{Pr(b_2|b_1), Pr(b_1|b_2)\}. \tag{16}$$

Let $j* = argmin_{j=1,2}\{supp(\{b_j\})\}$, we call item b_{j*} the *dependent node* and item $X\backslash\{b_{j*}\}$ the *master node* of b_{j*}. The dependence between these two items can be denoted by $< X\backslash\{b_{j*}\}, b_{j*} >$. Relation $\theta_X \to 1.0$ indicates that the dependent intensity of b_{j*} on item $X\backslash\{b_{j*}\}$ is almost 100%.

In this work, the *max-confidence* is used to measure the dependency of a candidate ToI on a *hot location*: given a predefined value of $\theta_0 \in [0, 1]$, we called $b \in \Sigma_{CUT(b^H)}$ a ToI associated to *hot location* b^H, if

$$\theta_{\{b^H, b\}} \geq \theta_0. \tag{17}$$

6.2.2. The threshold for max-confidence. Mutual information (MI) has been used to characterize both the relevance and redundancy of variables. Given a term b and category b^H, the mutual information criterion between b and b^H is defined to be [31]:

$$MI(b, b^H) = log\frac{Pr(b|b^H)}{Pr(b)}. \tag{18}$$

$MI(b, b^H)$ expresses the quantity of information that one can obtain about b by observing b^H. In general, the stronger the quantity, the greater the $MI(b, b^H)$ and vice versa.

Given a *hot location* b^H and a ToI of $b \in \Sigma_{CUT(b^H)}$, if $supp(\{b^H\}) \geq supp(\{b\})$ (the location is more b^H famous than its ToI), then we obtain the following result from relation (16):

$$\theta_{\{b^H, b\}} = \frac{supp(\{b^H b\})}{supp(\{b\})} = \frac{\sigma(\{b^H b\})}{\sigma(\{b\})} \geq \theta_0. \tag{19}$$

Taking the logarithm on both sides of relation (19), we can get

$$log\left\{\frac{Pr(b|b^H)}{Pr(b)}\cdot\frac{|CUT(b^H)|}{\theta_0|\boldsymbol{B}|}\right\}\geq 0. \tag{20}$$

From the perspective of mutual information, we obtain:

$$MI(b,b^H)-log\frac{\theta_0}{\frac{|CUT(b^H)|}{|\boldsymbol{B}|}}\geq 0\Rightarrow min\{MI(b,b^H)\}=log\frac{\theta_0}{\frac{|CUT(b^H)|}{|\boldsymbol{B}|}}. \tag{21}$$

The mutual information is always greater than or equal to zero, with equality if b and b^H are independent. To keep $MI(b, b^H) > 0$, we can expect that $min\{MI(b, b^H)\} \geq 0$ which means

$$\theta_0\geq\frac{|CUT(b^H)|}{|\boldsymbol{B}|}. \tag{22}$$

On one hand, similar to other metrics used in data mining tasks, the value of θ_0 can be set as any value between 0 and 1 theoretically. On the other hand, relation (22) provides a benchmark to evaluate the closeness of extracted ToI to a *hot location*: to extract the exact ToI for a *hot location* (b^H) with a bigger $CUT(b^H)$, we need a bigger value of θ_0. Accordingly, if we lose the constraints of θ_0 to obtain more ToI for a location, then more noise will be extracted.

Note that MI has some similar properties as *max-confidence*; however, it does not take the impact of word frequency into consideration. It therefore has inferior performance due to a bias favouring rare terms [31]. *Max-confidence* makes up this deficiency with the *minimum support* threshold in frequent pattern mining process. That is why *max-confidence* performances could be better than using MI directly in ToI extraction.

6.3. ToI extraction algorithm

Given a *hot location* $b^H \in FP^{(1)}(\boldsymbol{B}_G)$ and a potential ToI term b, the ToI extraction processes are mainly focused on calculating the value of $supp(b|CUT(b^H))$ and $min\{supp(b^H), supp(b)\}$ according to **Lemma 2 and Lemma 3**, and analysing the dependency of b on b^H in datasets with *max-confidence*.

The codes in **Algorithm 3** show the skeleton of our ToI extraction processes. As illustrated, it goes through the three phases:

- Finding out the *hot locations* mentioned in blog i (Line 5).
- Obtaining *cut vector* for b_{ij}^H in blog i and putting it into the appropriate dataset of $CUT(b^H)$ (Lines 6–10). Finally, all the $CUT(b^H)$ were put into CUT (Line 11).
- Analysing the dependency of all the elements in $\Sigma_{CUT(b^H)}$ on *hot location* b^H (Lines 12–24).

To obtain the complete interdependent relationships in $CUT(b^H)$, we set the max-confidence computation as a progressive process (Lines 16–22):

- First, we check the dependency relationship between any ToI candidate, b, with b^H, where $b \in FP^{(1)}(CUT(b^H))$. If $\theta_{X=\{b^H,\,b\}} > \theta_0$, then we obtain a dependency relationship as $< b^H, b >$.

Further, we check the dependency relationship between b and any other ToI candidate $\boldsymbol{b}'$, where $\{b, \boldsymbol{b}'\} \in FP^{(2)}(CUT(b^H))$. Similarly, if $\theta_{X=\{b,\,\boldsymbol{b}'\}} > \theta_0$, then we can obtain a transitive dependency relationship as $< b^H, b, \boldsymbol{b}' >$.

Note that: (1) the analysis of the dependency of potential ToI terms b on b^H in dataset $CUT(b^H)$ needs to scan all the items in $\boldsymbol{B}$; (2) according to **Lemma 3**, we use the results of $FP^{(1)}(CUT(b^H))$ to obtain the support value of any 2-itemset like $\{b^H, b\}$, which would simplify the frequent pattern mining process; and (3) the computation complexity has been approximate to $\#\ of\ hot\ terms \times |FP^{(1)}(CUT(b^H))| \times |FP^{(2)}(CUT(b^H))|$. That is, the efficiency of the algorithm is affected by the *minimum support* threshold in mining frequent 1- and 2-itemset from $CUT(b^H)$.

Algorithm 3. ToI extraction algorithm.

Input: Word vector set $\boldsymbol{B}$, Geographical word vector set $\boldsymbol{B}_G$, Hot tourism location set $\Sigma_{\mathbf{B}_{G^H}}$, θ_0;
Output: ToI set;
$CUT = \phi$;
for $i = 1$ to $|\boldsymbol{B}|$ **do**
  $\boldsymbol{b}_{G_i^H} = \boldsymbol{b}_{G_i} \cap FP^{(1)}(\boldsymbol{B}_G)$;
  for $j = 1$ to $|\boldsymbol{b}_{G_i^H}|$ **do**
    Calculate $CUT(b_{ij}^H)$ from $\boldsymbol{b}_i$;
    $CUT(b^H) \leftarrow CUT(b_{ij}^H)$ where $b^H = b_{ij}^H$;
  end for
end for
$CUT \leftarrow CUT(b^H)$ for all the $b^H \in FP^{(1)}(B_G)$;
Calculate $FP^{(1)}(B)$;
for $i = 1$ to $|CUT|$ **do**
  Calculate $FP^{(1)}(CUT_i)$ and $FP^{(2)}(CUT_i)$;
  for each $b \in FP^{(1)}(CUT_i)$ **do**
    if $\theta_{\{b_i^H, b\}} \geq \theta_0$ **then**
      if $\theta_{\{b,b'\} \in FP^{(2)}(CUT_i)} \geq \theta_0$ **then**
        $ToI(b_i^H) \leftarrow < b^H, b, \boldsymbol{b}' >$;
      else
        $ToI(b_i^H) \leftarrow < b^H, b >$;
      end if
    end if
  end for
end for
return $\cup ToI(b_i^H)$.

Table 7. Statistical description of the dataset used in the experiments.

Items	Statistical description
Number of users	359
Number of blogs presented	396
Total number of words	88,373
Average length of blog	2230 words
Number of nouns (ratio)	11,700 (57%)
Number of geographical terms	595

7. Experimental results

The presented method can be used to summarize topic-related information from massive text documents. In this section, we present an experiment study to demonstrate the proposed method.

7.1. Experiment setup

The blog data were extracted from www.mafengwo.com, one of the most famous blog sites in China for tourism information sharing. Altogether, 450 blogs posted from 1 October 2006 to 31 January 2014 in 'Hong Kong' (targeted destination) tourism channel were collected with a blog extraction tool. The blogs with empty text (some blogs are pictures only) were removed and 396 valid travel blogs were remained for the following experiments. Table 7 summarizes the dataset used in the experiments.

The contents of *GNT* are extracted from the attraction terms on www.tripadvisor.com. Note that the blogs were mostly written in Chinese. In the following, we do experiments with data in Chinese characters and report the results in English.

7.2. Word segmentation

At the initial stage of the experiments, the word segmentation method is introduced to tokenize the blog text into words so that we obtain an initial dataset of *all terms* after basic data cleaning. Then, we extract the dataset of *nouns* from the

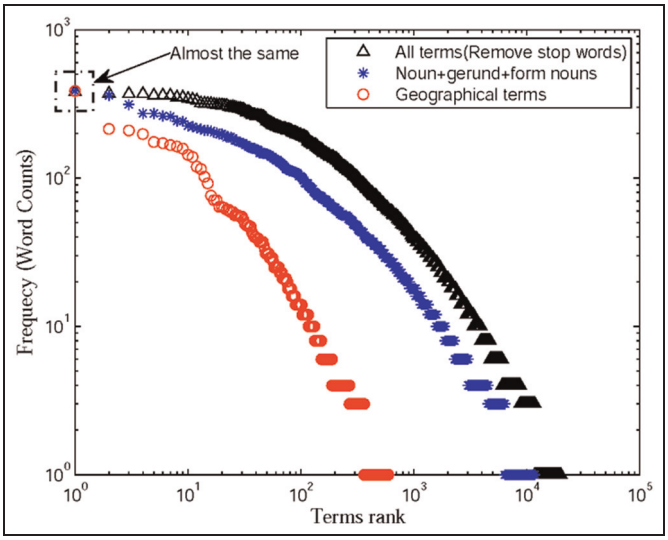


Figure 4. Frequencies of word segment in blogs of Hong Kong tourism.

all terms by removing the non-noun *word segments*. To identify the *hot locations* and their sequential relations efficiently, we further select a special dataset of *geographical terms* from the *nouns* according to the *GNT*. The frequency of terms in the dataset of *all terms, nouns* and *geographical terms* are sorted in Figure 4.

Different from the systems characterized by short text, such as micro-blogs [45], the frequency of terms in *all terms* to their corresponding ranks does not follow the common power-law distribution. The most likely reason is that the document length of blogs is much longer than that of the contents in BBS or microblogs. This result indicates that the blogging behaviours of bloggers are independent from each other, but they try to sketch out the travel experiences in detail to obtain blog readers' appreciations, which will result in longer document length and various (but partly common) words used in blog content. On the other hand, the ranks of geographical terms to their corresponding frequencies is of power-law distribution. This illustrates that few geographical terms are very popular while most of the others are not.

Moreover, the frequencies of the most popular (top ranked) terms in the dataset of *all terms, nouns* and *geographical terms* are almost the same (see the upper left corner in Figure 4). Obviously, they are geographical terms. It indicates that the geographical term plays a very important role in the content of the tourism blogs.

7.3. Hot tourism locations and travel routes

Intuitively, popular geographical terms in blogs represent the tourism locations that most people are concerned with [45], thus the top *frequent geographical terms* can be seen as the most popular tourism locations.

For the blogs about Hong Kong tourism, the ranks of geographical terms to their frequency are shown in Figure 5a. There are two significant turning points on the curve: '*Central*' and '*Kowloon*'. In order to reserve as much valuable information as possible, we take the top 15 geographical terms whose frequencies are bigger (and equal) than that of '*Kowloon*' as the *hot locations* in Hong Kong.

By using the 15 *hot locations* as network nodes and analysing their position relations in the blogs, we can sketch out a popular travel routes for bloggers travelling in Hong Kong (Figure 5b). Each node represents a *hot location*, and the node size shows the frequency of the location in all the blogs. Obviously, the larger the node, the tour location it represented is more frequent. The lines in Figure 5b show that some travel sequential relationships between the connected nodes exist. For example, 'Ocean park-Disneyland', 'Central-Harbour City' and so on.

Note that the most popular term, i.e. the term 'Hong Kong', performs a hierarchical relationship with the other terms. We keep these relations to show that the other terms are ToI for 'Hong Kong' at a macro level. In other words, they are the common knowledge of 'what to play' when people go to 'Hong Kong'.

7.4. ToI extraction

We use '*Disney land*' as an illustration for the ToI extraction. First, the local features located near the term 'Disney land' and bound by a pair of adjacent punctuation marks are cut from each blog generated *word vector* to form the dataset of

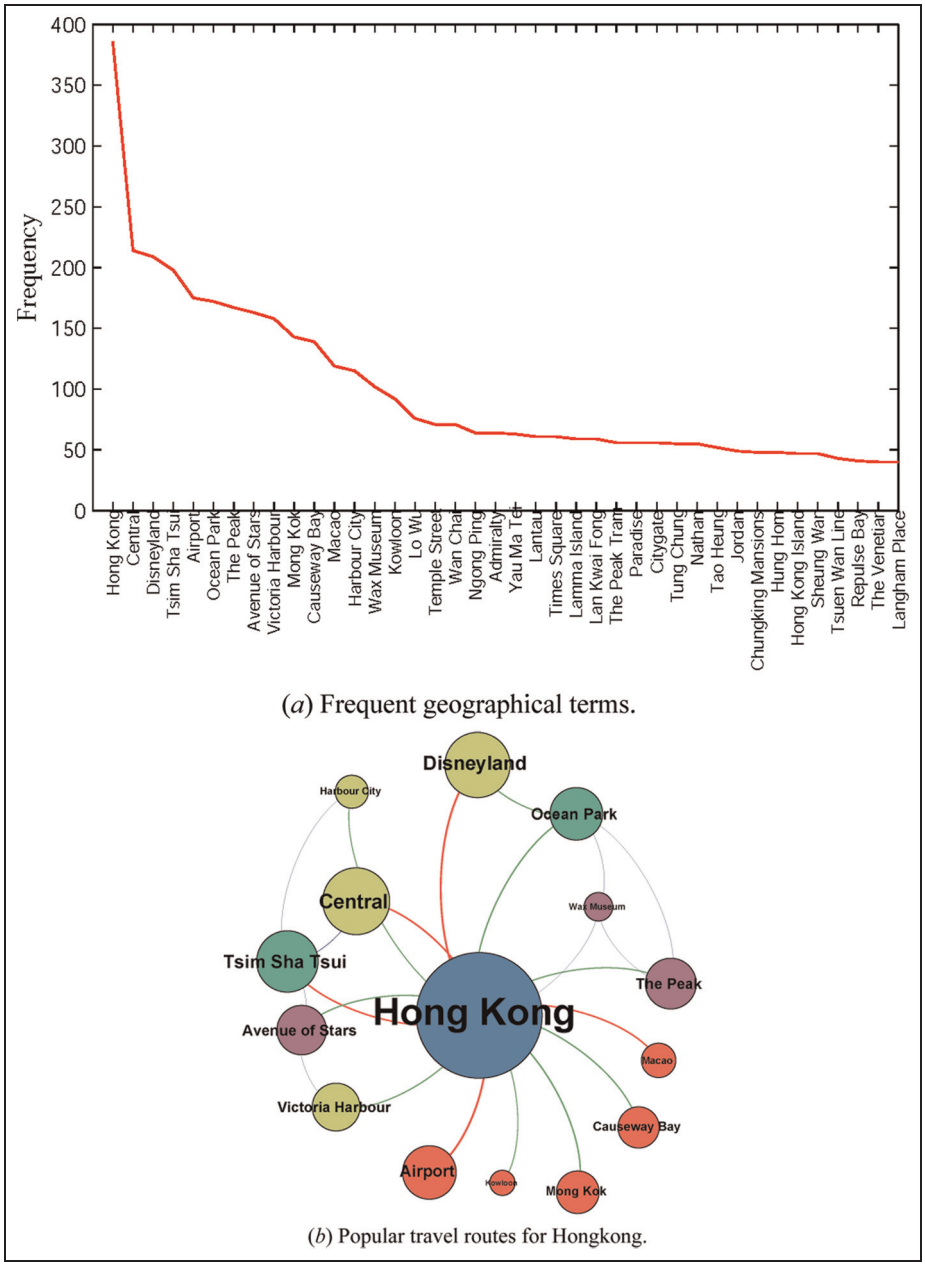
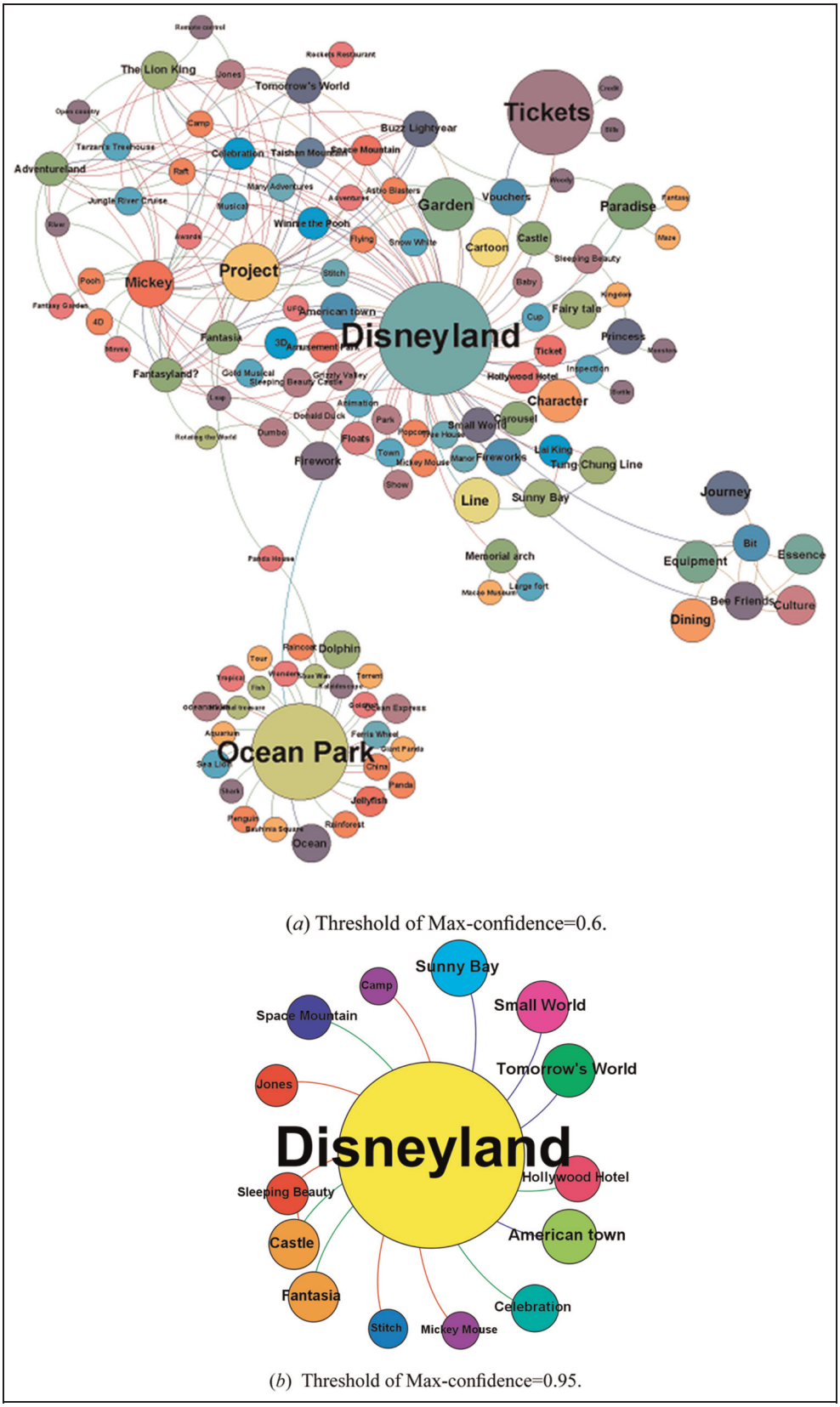


Figure 5. Backbone-nodes-based travel routes for Hong Kong.

CUT('*Disney land*'). Then, the frequent 1-itemsets and 2-itemsets in *CUT*('*Disney land*') are mined. Finally, the extracted ToI for the *hot location* of 'Disney land' are shown in Figure 6.

By setting the threshold of *max-confidence* as a more relaxed value of 0.6, people can obtain more ToI around Disneyland in Hong Kong (Figure 6a). Some nearby tourist attractions, such as '*Ocean Park*', and their primary ToI have also been extracted partially because they are described frequently by bloggers in the same context (i.e. the same *cut vector*) of a comparative or associated manner. Interestingly, other matters related to Disney tourism, such as the ticket ('*Tickets*') and public transport ('*Tung Chung Line*') are also presented, which are ToI highly relevant to Disney tourism and may provide richer information for people's travel planning. However, a relatively loose threshold of *max-confidence* would cause the correlations between ToI to become more complex. To obtain the clear relationship between a *hot location* and its most close ToI, thus, a bigger threshold of *max-confidence* is needed (Figure 6b). As we can see, these ToI are the most popular tourist projects for '*Disney land*'.



(*a*) Threshold of Max-confidence=0.6.

(*b*) Threshold of Max-confidence=0.95.

Figure 6. Extracted ToI for 'Disney park'.

Table 8. Reference threshold of *max-confidence* for the top 14 *hot locations*.

Hot location b^H	$\|CUT(b^H)\|$	Reference value of θ_0
Central	213	0.54
Disneyland	209	0.53
Tsim Sha Tsui	197	0.50
Airport	175	0.44
Ocean Park	172	0.43
The Peak	166	0.42
Avenue of Stars	162	0.41
Victoria Harbour	157	0.40
Mong Kok	143	0.36
Causeway Bay	138	0.35
Macao	119	0.30
Harbour City	114	0.29
Wax Museum	101	0.26
Kowloon	91	0.23

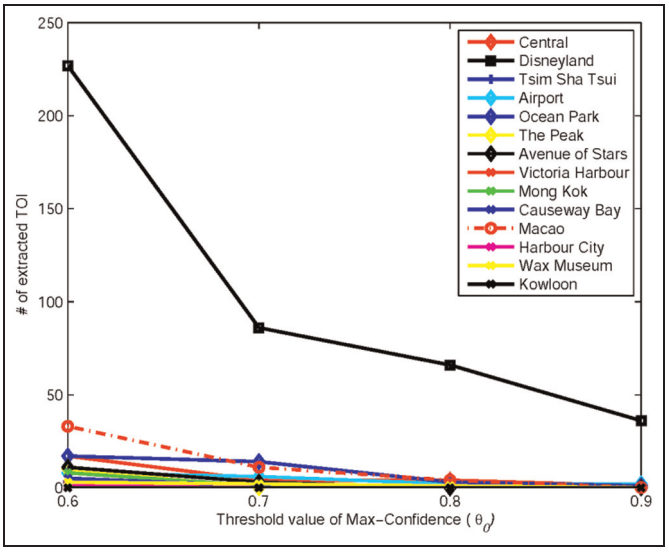


Figure 7. Number of extracted ToI for the top 14 *hot locations*.

The subset size and the reference value of θ_0 for the top 14 *hot locations* are presented in Table 8. We can see that the more popular the location is, the higher value of θ_0 is needed for its ToI capturing. This is rational since the popular locations would be mentioned by bloggers in various context. So, in the information mining task, a smaller value of θ_0 would cause the infrequent features and noise are also associated with hot words.

At different values of θ_0, the number of extracted ToI for the top 14 *hot locations* in Hong Kong (term 'Hong Kong' is ignored) are shown in Figure 7, in which, we can see that 'Disneyland', 'Macao' and 'Ocean park' are three *hot locations* having more ToI than the others, implying that when people tour in these locations, they may spend more time and money. This is in accordance with common sense in nature. For the rest *hot locations* with few ToI, people can bind them into several travel packages according to their geographic relationship so that these locations can be visit together in the same package.

7.5. Performance

In this section we discuss the measures used in evaluating the performance of the experiments as well as the parameter settings.

Table 9. Classification of the possible results of a ToI extraction task.

	Extracted	Not extracted
ToI relevant to the *hot location*	True-positive (*tp*)	False-negative (*fn*)
ToI irrelevant to the *hot location*	False-positive (*fp*)	True-negative (*tn*)

7.5.1. Evaluation measures. For the task of blog extracting, if we select a test hot location and ask an extraction method to find out its ToI, eventually we have four possible outcomes for the extracted and inherent ToI, as shown in Table 9.

In literature of information retrieval, two terms of *Precision* and *Recall* are commonly used to measure the extraction results:

$$Precision = \frac{\#tp}{\#tp + \#fp} \quad \text{and} \quad Recall = \frac{\#tp}{\#tp + \#fn}. \tag{23}$$

To facilitate calculation, we define two types of noises in the extracted contents by a method:

- $Noise_1$ is the common thing, i.e. things that can be seen somewhere else. For example, 'street', 'subway', 'shop', etc.
- $Noise_2$ is the other *hot locations*, i.e. we avoid to use one *hot location* as a ToI of another *hot location*.

Hence, given an extraction method, we calculate the value of *#tp* as:

$$\#tp = \#\ Extracted\ Items - \Sigma_{i=1,2} \#\ Noise_i. \tag{24}$$

Another problem in performance evaluation is the lack of ground truth for the exact number of ToI, i.e. $\#tp + \#fn$, for a *hot location*. However, we see that the value of $\#tp + \#fn$ is the same in any method, which enables us to obtain a more simple result of $Recall \propto \#tp$.

In the following, we will compare the efficiency of our method, namely *Term Vector Subdividing* (TVS), with some classic methods, and show the comparison results of (1) the average *precision* and (2) the average number of ToI (*#tp*) in the extracted top-k terms.

7.5.2. Experimental results. First, we conduct the comparison experiment with TVS and LDA on the dataset of *nouns*. The results are shown in Figure 8. Notably, in extracting a small number of ToI (e.g. less than 50), the precision of TVS is superior to that of LDA (Figure 8a). This shows that the TVS method is good at extracting terms that link very closely with the key term (*hot location*). However, the accuracy of TVS begins to decrease with the increasing number of top-k threshold (method is required to provide more ToI), whereas LDA performs better. One possible explanation is that TVS must run with a relatively lower threshold of *max-confidence* when it is required to provide more extraction contents. Obviously, this may introduce increasing number of noise into the ToI candidates.

In addition, TVS is a method based on feature selection with the metric of *max-confidence*. Hence, we conduct some experiments to see the differences between the classic TF-IDF, DF and the *max-confidence* in ToI extraction (the comparison between MI and *max-confidence* is discussed latter)[3]. The dataset used here are $CUT(b_i^H)$, where b_i^H $(i = 14)$ is one of the top 14 *hot locations* in Hong Kong (Figure 8b). The average results are shown in Figure 9:

- For the overall results, *max-confidence* dominated all other metrics, and the DF metric shows the worst performance. *Max-confidence* is better than TF-IDF because TF-IDF does not take into account interesting word co-occurrences containing terms with low IDF [46].
- Along with the number of requested features becoming bigger, the number of extracted ToI by different metrics all keep increasing (Figure 9b). This result makes sense that a bigger threshold will result in a larger set of candidates for real ToI.

7.5.3. Semantic comparison of the extracted ToI. We are interested to see what the differences are in semantics for the extracted contents by using the different methods. Table 10 lists the top five contents extracted by LDA, TF-IDF-based

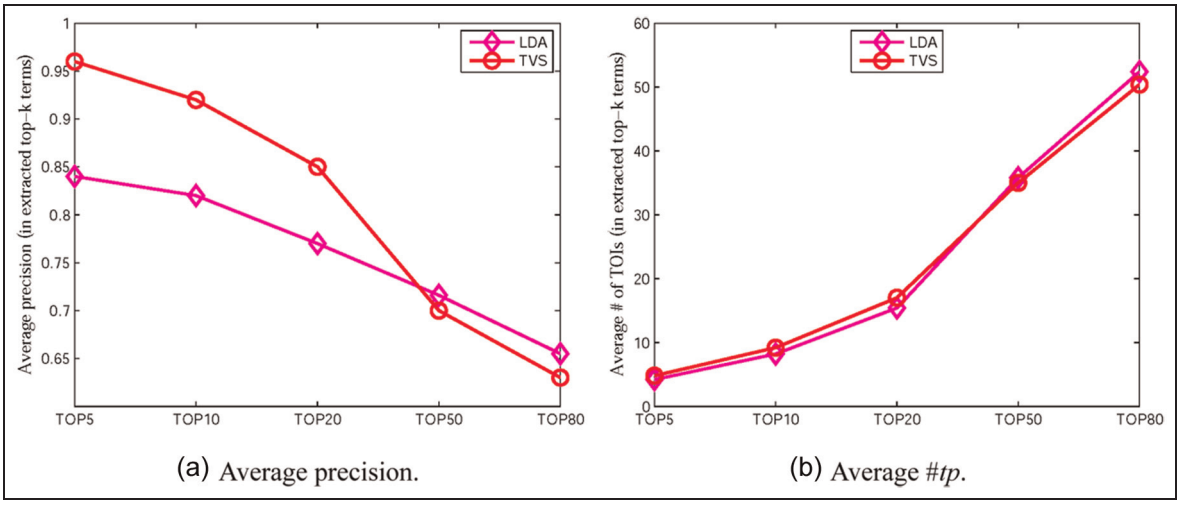


Figure 8. Comparison results between TVS and LDA.

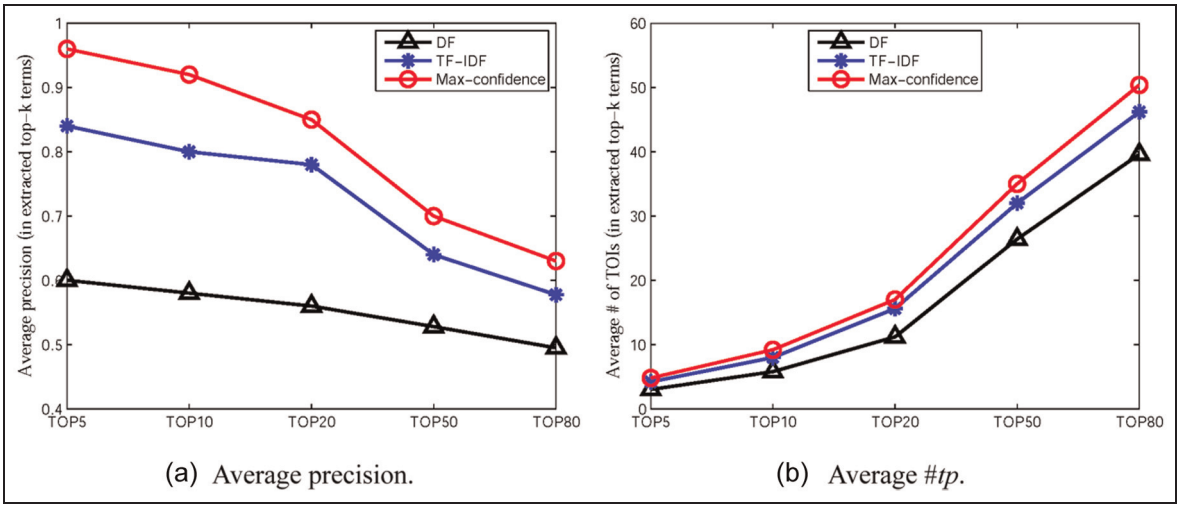


Figure 9. Comparison results between metrics.

method and TVS for 10 hot tourism locations in Hong Kong. As we can see, LDA and TF-IDF tend to extract the adjacent *hot locations* (in bold) and the hot common things (underlined) as the ToI for a given location, whereas TVS would provide people with local ToI which have very close relationships with the *hot location*. Interestingly:

- There are few common terms extracted by the three methods. Obviously, the mechanisms of these methods are different from each other.
- Based on common sense, 'Mickey' should be a strong local ToI for 'Disneyland'. However, it does not appear in the top five results extracted by TVS (it appears in the top 10 results). One possible reason for that is 'Mickey' is so hot in 'Hong Kong' that many bloggers would mention it outside of the context of 'Disney'.

8. Discussion

As a text mining related work, two general tasks of algorithm scalability and sentiment analysis is discussed as follows.

Table 10. The extracted top five contents of the LDA, TF-IDF and TVS.

Hot Location	TF-IDF	LDA	TVS
Central	Star Ferry	Hong Kong	Queen's Road
	Pier	**Victoria Harbour**	Garden Road
	Wan Chai	**Avenue of Stars**	International Finance Centre
	Causeway Bay	**Tsim Sha Tsui**	Yung Shue Wan
	Sheung Wan	Star Ferry	Cheung Chau
Disneyland	Mickey	Time	Sunny Bay
	Ocean Park	Project	Small World
	Tickets	People	Toy Story (Playland)
	Sunny Bay	Paradise	Sleeping Beauty
	Special-Line	Garden	Stitch
Tsim Sha Tsui	Yau Ma Tei	**Mong Kok**	Granville
	Harbour City	**Causeway Bay**	Canton Road
	Star Ferry	**Central**	Tsim Sha Tsui East
	Kowloon	Subway	Star Ferry
	Temple Street	Shop	Yau Ma Tei
Airport	Airport Express Line	Aircraft	Baoan International Airport
	Aircraft	Hong Kong	Boarding Pass
	Flight	Time	Chek Lap Kok International Airport
	Duty Free	Weather	Airport Express Line
	International Airport	Aviation	Duty Free
Ocean Park	**Wax Museum**	Park	Ocean Express
	Disneyland	Ocean	Ferris Wheel
	Tickets	World	Chunghwa
	Admiralty	Dolphin	Jellyfish
	Ocean	Show	Aquarium
The Peak	**Wax Museum**	Tram	Sky Terrace
	Tram	People	The Sweet Dynasty
	Sky Terrace	Hilltop	**Wax Museum**
	Night view	Ngong Ping	Garden Road
	Ocean Park	**Wax Museum**	Wong Tai Sin Temple
Avenue of Stars	**Victoria Harbour**	Hong Kong	Bruce Lee
	Wax Museum	**Victoria Harbour**	Handprints
	Ruins	**Central**	McDull
	Star Ferry	**Tsim Sha Tsui**	Museum of Art
	Golden Bauhinia Square	Star Ferry	Civil Administration
Victoria Harbour	**Avenue of Stars**	Hong Kong	Two banks of Victoria Harbour
	Star Ferry	**Central**	Night Tour
	Night View	**Avenue of Stars**	Golden Bauhinia Square
	Golden Bauhinia Square	**Tsim Sha Tsui**	Museum of Art
	Wax Museum	Star Ferry	Fortaleza do Monte
Mong Kok	Yau Ma Tei	**Causeway Bay**	Prince Edward Station
	Sneaker Street	**Tsim Sha Tsui**	Sai Yeung Choi Street
	Ladies Street	**Central**	Sneaker Street
	Langham Place Hotel	Subway	Tung Choi Street
	Nathan Road	Street	Garden Street
Causeway Bay	Times Square	**Mong Kok**	Hysan Plaza
	Wan Chai	**Tsim Sha Tsui**	Lockhart Road
	SOGO	**Central**	Hennessy Road
	Sheung Wan	Subway	Times Square
	Central	Shop	SOGO

8.1. Scalability of the method

For the scalability issue, we think it is also a question of the method performance on big data. Therefore, we address this point from two aspects: algorithm's capability of parallelization and algorithm's performance on a single computer.

8.1.1. Analysis of the algorithm's capability of parallelization. For the task of blog extracting, if we select a test hot location and ask for an extraction method to find out its ToI, eventually we have four possible outcomes for the extracted and inherent ToI, as shown in Table 9.

Recently, the MapReduce framework [47], a popular programming model for processing and generating large datasets with a parallel, distributed algorithm on a cluster (constructed with a group of servers), has drawn much attention and been widely used in both academia and industry on large scale data processing.

The core algorithms in this work are *word pre-processing, hot tourism location mining* and *ToI extraction*. Fortunately, they can be implemented easily on a MapReduce environment to perform the mining tasks on a very large dataset.

- For the *word pre-processing* process, first the web page crawling task can be conducted in parallel. Then, the crawled dataset of documents can be separated into smaller parts and each part assigned to a single server (Mapping), on which the word segmentation and stop word removing task can be run efficiently. The result is to generate a set of word vectors.
- For the *hot tourism location mining* process, the hot tourism location mining task is a process of identifying hot tourism terms; it also can be easily conducted on a MapReduce framework.
- For the *ToI extraction* process, the mapping process can be implemented as follows. First, divide the geographical word vector set into smaller parts and assign each of the parts to a server. Then, find $CUT_i(b^H)$ in server i for hot word b^H. The reducing process is simply to combine $CUT_i(b^H)$ into a single $CUT(b^H)$.

8.1.2. Experiments with the algorithm's performance on a single computer. We have conducted some scalability and sensitivity related experiments for this part using a single computer. The computer is a Lenovo Thinkpad X230i with an Intel CPU (Core i5 2410M) and 4G RAM running on the Windows 7 system. The software is programmed and implemented with Python.

- Influence of document number (quantity) on efficiency

The primary experimental results are shown in Figure 10, in which we can see that the running time is increasing almost linearly along with the increasing of the document number. For the most time-consuming of computation experiments (#document=10,000, which formed a data file of 76.3 M initially, and a file of 84.3 M after word segmentation), the computation will be finished efficiently with 5 min. It can be expected that the time consumption will be controlled easily even as the #document rises to relatively huge levels of millions or more.

- Influence of minimum support on efficiency

Frequent patterns (FPs) are itemsets, subsequences or substructures that appear in a dataset with frequency no less than a user-specified threshold of minimum support (min_supp). In this work, frequent itemsets play an essential role that try

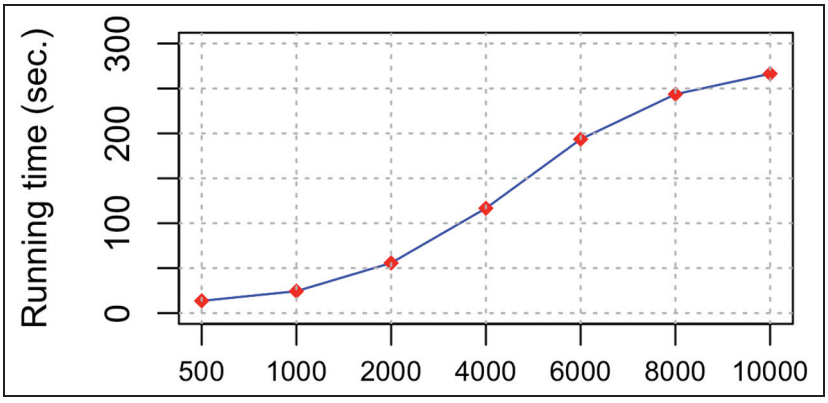


Figure 10. Influence of document quantity on efficiency.

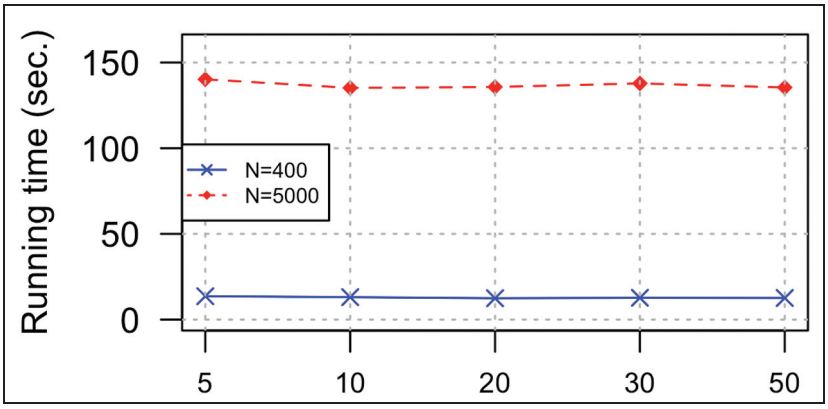


Figure 11. Influence of support count on efficiency.

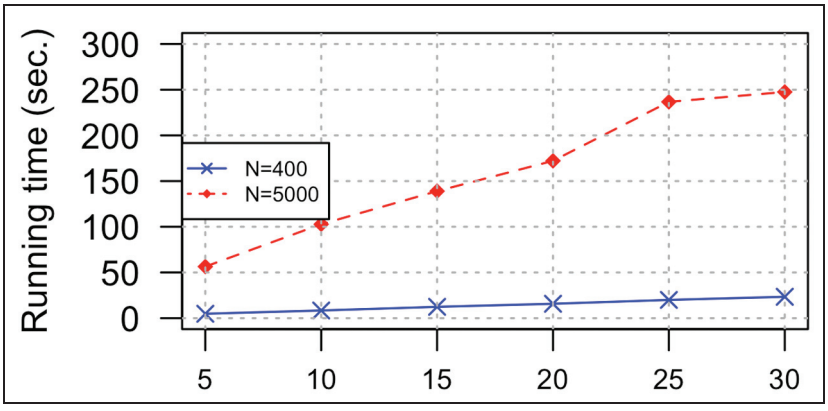


Figure 12. Influence of the recommended number of hot locations on efficiency.

to find interesting patterns from databases. However, in frequent pattern mining, the important parameter of minimum support should be specified properly. It is well-known that setting a uniform threshold for the entire data makes it hard to get sufficiently interesting information: support that is too low may introduce noisy patterns, and support that is too high discards even useful patterns.

We performed a set of experiments on two datasets of 400 and 5000 documents to explore the influence of the threshold of min_supp (support count ranged from 5 to 50). The final results are presented in Figure 11. The time costs are relatively stable at the same level of min_supp. Once again, it proves that the running time is greatly influenced by the number of articles (N). For the dataset of 5000 documents, the running efficiency is about 135 s (2.3 min), which shows that the proposed method can perform efficiently on mining frequent correlations from a large scale dataset with a relatively small threshold (more tourism features will be presented).

- Influence of the recommended number of hot locations

For any BI system, if it cannot report the complete set of the mined hot locations, top-k is a best choice in final results presentation [48]. Also, the number of k will have an impact on the algorithm performance.

We have conducted a set of experiments on two datasets of 400 and 5000 documents to explore the effects of value k (ranging from 5 to 30) on algorithm performance. The time cost was increased linearly along with the increasing value of k (Figure 12). The more information a user wants to mine from online documents, the more time he will spend on data

Table 11. Summary of negative sentiments for the top 14 locations.

Initial Rank	Hot location	Negative reviews (%)	New rank (Negative reviews) removed)
1	Central	26.89	3
2	Disneyland	26.51	1
3	Tsim Sha Tsui	23.53	2
4	Airport	22.86	5
5	Ocean Park	22.67	6
6	The Peak	18.67	4
7	Avenue of Stars	17.90	7
8	Victoria Harbour	17.20	8
9	Mong Kok	23.08	9
10	Causeway Bay	25.36	10
11	Macao	22.69	12
12	Harbour City	15.79	11
13	Wax Museum	18.18	13
14	Kowloon	19.78	14

processing. Similarly, the most time-consuming task in these experiments is about 4 min which has the intent of recommending the top 30 hot locations. This result illustrates that the proposed method in this work has less pressure with the top k recommendation strategy in text mining tasks.

8.1.3. Summarization. Both discussions about the algorithm's capability of parallelization and experimental performance on a single computer show that the proposed method is efficient in situations where scalable computation is needed.

8.2. Sentiment analysis issue

Previous studies in literature have shown that it is a careful selection and evaluation process for people to make a tourism plan [18]. The tourism experiences under such a plan were mostly satisfactory. In general, this would result in motivating the sharing of pleasant experiences rather than complaining when contributing to an online travel blog [49, 50].

It would therefore be expected that the negative sentiment would have little impact on hot location mining. Along this line, in the following Table 11, we summarize the percentage of negative reviews for the top 14 locations. As we can see, the average percentage of negative reviews is about 21.51%.

Further, we removed the negative reviews and recalculated the frequency of the terms of the top 14 locations; the new rank is also shown in Table 11. The data show that negative emotion has little effect on hot location identification. Note that under the situation of tourism recommendation, the bloggers' sentiments for a location may have heavy impact. That is another challenge.

On the other hand, the sentiments published online may encounter the evaluation bias problem due to the various backgrounds and reasons of the bloggers, for example, personal preferences and some unpleasant incidents in travel. Thus, in this work, our main intention is to find a set of hot locations as well as their local features in a targeted city for readers.

9. Conclusion

Online tourism blogs have become an important resource for information sharing because they can provide valuable information for new customers from many experienced users. However, exploring useful information from such massive data on the Internet will lead to problems with information overload and noise disturbance.

In this work, we propose a research methodology to summarize the popular information from massive tourism blog data. To this end, we first crawl the blog contents from the website and divide them into semantic word vectors as data source. Second, we collect the geographical data from all the blog vectors, and mine the hot tourism locations and their frequent sequential relations in it. The results of this part can be used to summarize the popular information about 'where to go' (trip route) in a set of tourism blogs. We then propose a vector subdividing method to collect local features for each hot location, and introduce the max-confidence metric to identify the ToI for the corresponding hot location. The captured ToI for each hot location are accounts for the question about 'what to play' at a specific tourist location. Notably,

the significant result of this method is that the disturbances from high frequent irrelevant word (noise) are shielded. Finally, we illustrate the benefits of this approach by applying it to a Chinese online tourism blog dataset.

The experiment results show that the proposed method can be used to explore the hot tourism locations (and their frequent travel sequences as well) and their corresponding ToI from massive blogs efficiently. Future work will be to reduce the algorithm complexity and present an optimization method to extract more precise correlations for a hot term.

Acknowledgements

The authors thank the anonymous reviewers for their thoughtful comments and suggestions.

Funding

This work was partly supported by the National Natural Science Foundation of China (No. 71402007 / 71271044 / U1233118 / 71572029) and the Fundamental Research Funds for the Central Universities (No. 2014RC0601).

Notes

1. Extracted from www.aluxurytravelblog.com.
2. If b_{ij} is a punctuation mark, then $b_{ij} =$ '||'.
3. IG and χ^2 are not suitable for the experiments due to the *cut vector* merging operation.

References

[1] Werthner H and Ricci F. E-commerce and tourism. *Communications of the ACM* 2004; 47(12): 101–105.
[2] Pang B and Lee L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2008; 2(1–2): 1–135.
[3] Asbagh M, Sayyadi M and Abolhassani H. Blog summarization for blog mining. In: Lee R and Ishii N (eds) *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. Berlin: Springer, 2009, pp. 157–167.
[4] Salton G, Wong A and Yang CS. A vector space model for automatic indexing. *Communication of the ACM* 1975; 18(11): 613–620.
[5] Soucy P and Mineau GW. Beyond TFIDF weighting for text categorization in the vector space model. In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05)*, 2005, pp. 1130–1135.
[6] Turney PD and Pantel P. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 2010; 37(1): 141–188.
[7] Robertson S. Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation* 2004; 60(5): 503–520.
[8] Yang Y and Wilbur JW. Using corpus statistics to remove redundant words in text categorization. *Journal of the American Society for Information Science* 1996; 47(5): 357–369.
[9] Qamra A, Tseng BL and Chang EY. Mining blog stories using community-based and temporal clustering. In: *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, 2006, pp. 58–67.
[10] Attardi G and Simi M. Blog mining through opinionated words. In: *Proceedings of the Fifteenth Text REtrieval Conference*, 2006, pp. 14–17.
[11] Cao Q, Duan W and Gan Q. Exploring determinants of voting for the ''helpfulness'' of online user reviews: A text mining approach. *Decision Support Systems* 2011; 50(2): 511–521.
[12] Ghose A and Ipeirotis PG. Estimating the helpfulness and economic impact of product reviews: mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering* 2011; 23(10): 1498–1512.
[13] Hu M and Liu B. Mining and summarizing customer reviews. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 168–177.
[14] Dave K, Lawrence S and Pennock DM. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: *Proceedings of the 12th international conference on World Wide Web*, 2003, pp. 519–528.
[15] O'Leary DE. Blog mining-review and extensions: ''From each according to his opinion''. *Decision Support Systems* 2011; 51(4): 821–830.
[16] Li N and Wu DD. Using text mining and sentiment analysis for online forums hotspot detection and forecast. *Decision Support Systems* 2010; 48(2): 354–368.
[17] Pan B, MacLaurin T and Crotts JC. Travel blogs and the implications for destination marketing. *Journal of Travel Research* 2007; 46(1): 35–45.
[18] Sharda N and Ponnada M. Tourism Blog Visualizer for better tour planning. *Journal of Vacation Marketing* 2008; 14(2): 157–167.

[19] Hofmann T. Probabilistic latent semantic indexing. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 50–57.
[20] Blei DM, Ng AY and Jordan MI. Latent dirichlet allocation. *Journal of Machine Learning Research* 2003; 3993–1022.
[21] Banerjee A and Basu S. Topic models over text streams: A study of batch and online unsupervised learning. In: *SIAM International Conference on Data Mining*, 2007, pp. 431–436.
[22] Moghaddam S and Ester M. On the design of LDA models for aspect-based opinion mining. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 803–812.
[23] Kim HD, Park DH, Lu Y and Zhai CX. Enriching text representation with frequent pattern mining for probabilistic topic modeling. *Proceedings of the American Society for Information Science and Technology* 2012; 49(1): 1–10.
[24] Rokaya M, Atlam E-s, Fuketa M, Dorji TC and Aoe J. Ranking of field association terms using Co-word analysis. *Information Processing and Management* 2008; 44(2): 738–755.
[25] Figueiredo F, Rocha LCd, Couto T, Salles T, Goncalves MA and Meira W Jr. Word co-occurrence features for text classification. *Information Systems* 2011; 36(5): 843–858.
[26] Liu T, Liu S, Chen Z and Ma W-Y. An evaluation on feature selection for text clustering. In: *Proceedings of the Twentieth International Conference on Machine Learning (ICML'2003)*, 2003, pp. 488–495.
[27] Yang Y and Liu X. A re-examination of text categorization methods. In: *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 42–49.
[28] Joho H and Sanderson M. Document frequency and term specificity. In: *Large-Scale Semantic Access to Content (Text, Image, Video and Sound) Conference*, 2007, pp. 350–359.
[29] Sebastiani F. Machine learning in automated text categorization. *ACM Computing Surveys* 2002; 34(1): 1–47.
[30] Lee C and Lee GG. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing and Management* 2006; 42(1): 155–165.
[31] Yang Y and Pedersen JO. A comparative study on feature selection in text categorization. In: *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997, pp. 412–420.
[32] Peng H, Long F and Ding CHQ. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2005; 27(8): 1226–1238.
[33] Ko Y, Park J and Seo J. Improving Text categorization using the importance of sentences. *Information Processing and Management* 2004; 40(1): 65–79.
[34] Ng HT, Goh WB and Low KL. Feature selection, perceptron learning, and a usability case study for text categorization. *SIGIR Forum* 1997; 31(SI): 67–73.
[35] Gao J, Wu A, Li M and Huang C-N. Chinese word segmentation and named entity recognition: a pragmatic approach. *Computational Linguistics* 2005; 31(4): 531–574.
[36] Stavrianou A, Andritsos P and Nicoloyannis N. Overview and semantic issues of text mining. *Sigmod Record* 2007; 36(3): 23–34.
[37] Sproat R, Shih C, Gale W and Chang N. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics* 1996; 22(3): 377–404.
[38] Tang J, Li H, Cao Y and Tang Z. Email data cleaning. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2005, pp. 489–498.
[39] Tan P-n, Steinbach M and Kumar V. *Introduction to Data Mining*. Boston, MA: Addison-Wesley, 2005.
[40] Ng HT and Lee HB. Integrating multiple knowledge sources to disambiguate word sense: an exemplar-based approach. In: *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, 1996, pp. 40–47.
[41] Lin D. Using syntactic dependency as local context to resolve word sense ambiguity. In: *Proceedings of the Eighth Conference on European Chapter of the Association for Computational Linguistics*, 1997, pp. 64–71.
[42] Xiong H, Tan P-n and Kumar V. Hyperclique pattern discovery. *Data Mining and Knowledge Discovery* 2006; 13(2): 219–242.
[43] Jo Y and Oh AH. Aspect and sentiment unification model for online review analysis. In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, 2011, pp. 815–824.
[44] Wu T, Chen Y and Han J. Re-examination of interestingness measures in pattern mining: a unified framework. *Data Mining and Knowledge Discovery* 2010; 21(3): 371–397.
[45] Cui A, Zhang M, Liu Y, Ma S and Zhang K. Discover breaking events with popular hashtags in twitter. In: *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 1794–1798.
[46] Pons-porrata A, Berlanga-Llavori R and Ruiz-shulcloper J. Topic discovery based on text mining techniques. *Information Processing and Management* 2007; 43(3): 752–768.
[47] Dean J and Ghemawat S. MapReduce: simplified data processing on large clusters. *Communications of the ACM* 2008; 51(1): 107–113.
[48] Deshpande M and Karypis G. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)* 2004; 22(1): 143–177.
[49] Yoo KH and Gretzel U. What motivates consumers to write online travel reviews? *Information Technology & Tourism* 2008; 10(4): 283–295.
[50] Gretzel U, Yoo KH and Purifoy M. Online travel review study: Role and impact of online travel reviews, www.tripadvisor.com/pdfs/OnlineTravelReviewReport.pdf (2007, accessed 20 August 2015).