Article

Spatial Keyword Query of Region-Of-Interest Based on the Distributed Representation of Point-Of-Interest

Xiangdian Zhu, Ye Wu *, Luo Chen and Ning Jing

College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; zhuxiangdian13@nudt.edu.cn (X.Z.); luochen@nudt.edu.cn (L.C.); ningjing@nudt.edu.cn (N.J.)
* Correspondence: yewugfkdl@nudt.edu.cn; Tel.: +86-186-8499-6060

Received: 21 April 2019; Accepted: 19 June 2019; Published: 20 June 2019

Abstract: The tremendous advance in information technology has promoted the rapid development of location-based services (LBSs), which play an indispensable role in people’s daily lives. Compared with a traditional LBS based on Point-Of-Interest (POI), which is an isolated location point, an increasing number of demands have concentrated on Region-Of-Interest (ROI) exploration, i.e., geographic regions that contain many POIs and express rich environmental information. The intention behind the POI is to search the geographical regions related to the user’s requirements, which contain some spatial objects, such as POIs and have certain environmental characteristics. In order to achieve effective ROI exploration, we propose an ROI top-k keyword query method that considers the environmental information of the regions. Specifically, the Word2Vec model has been introduced to achieve the distributed representation of POIs and capture their environmental semantics, which are then leveraged to describe the environmental characteristic information of the candidate ROI. Given a keyword query, different query patterns are designed to measure the similarities between the query keyword and the candidate ROIs to find the k candidate ROIs that are most relevant to the query. In the verification step, an evaluation criterion has been developed to test the effectiveness of the distributed representations of POIs. Finally, after generating the POI vectors in high quality, we validated the performance of the proposed ROI top-k query on a large-scale real-life dataset where the experimental results demonstrated the effectiveness of our proposals.

Keywords: ROI exploration; spatial keyword search; distributed representation; environment semantics; deep learning

1. Introduction

Recent years have witnessed the rapid development of Internet technologies and sensor devices, which, in turn, has resulted in the explosive growth of geo-related information. According to the statistics, 18.78% of web resources contain geographic location information and 18.6% of information retrieval is related to location [1]. The current research focus is spatiotemporal data mining and pattern discovery via multi-source geospatial big data. The corresponding achievements are widely used in urban computing, social analysis, environmental monitoring, and other fields, which greatly improve the quality of people’s life [2,3]. As an important research direction, location-based services (LBSs) have attracted great attention from both the academic and industrial communities. A valuable research problem in this field is exploring the locations of interest in the city by mining geo-tagged data. However, most of the current location services only concentrate on the search of isolated locations, such as Point-Of-Interest (POI) queries and ignore the user’s demand for Region-Of-Interest (ROI) exploration. In many real-life application scenarios, it is common for users to find multi-functional ROIs related to their requests, e.g., a person is hoping to watch a movie in a nearby cinema after
drinking coffee in a cafe, which is a difficult question for traditional POI based methods to answer. In addition, research on ROI exploration can not only create a better service in application, but also create high research value in urban function analysis. It is worth noting that we used the definition of geographic regions [4] rather than image areas [5] for the ROI in our article.

At present, the existing ROI exploration methods are mainly based on the statistical information or the density information of the query elements [6–9], such as POIs with certain keywords [10,11], while neglecting the influence of internal and environmental characteristics of the region. Theoretically, judging how much the ROI is related to a query should take into consideration the distribution and type characteristics of all geographic objects in the ROI. For instance, the regional ecology decides the association between the ROI and query requirements. However, it is still quite challenging to measure the relevance between each spatial object and the query requirements [12]. Moreover, spatial-keyword queries with multiple query elements still remain difficult to cope with [11], and they are usually more complex and result in a high time consumption.

Bao J et al. [13] revealed that there was a strong correlation between the spatial distribution of geographic objects and their categories, indicating that spatial objects could be classified by their neighbors. Consequently, inspired by [14], we utilized a distributed representation model, i.e., the Word2Vec model, to explore the spatial distribution characteristics of geographic objects and the relatedness between their types. Specifically, this model projects words into a distributed space in accordance with its context in the document to capture the spatial distribution features and environmental information of each type of spatial object, such as the POI. Thus, the spatial object can be transferred into a high-dimensional vector, which encodes the association characteristics between the POI types. Then, in order to achieve an efficient query of the relevant ROI, we executed a grid division on the whole research region so that each grid contained a certain number of POIs. With the grids viewed as the candidate ROIs, the ROI vectors were obtained by their internal POI vectors, which implies the type characteristics and spatial distribution of each candidate ROI. Finally, after designing different query patterns, the similarity score between each ROI vector and the vectorized query conditions was calculated to find the top-K ROIs related to the query.

The contributions can be summarized as follows:

- Unlike traditional ROI exploration based on the statistical information of query elements, we studied ROI exploration for environmental semantics and distribution characteristics, which introduced new opportunities and challenges. We were the first to utilize the spatial distribution and the semantic information of the POI in ROI exploration.
- We proposed a novel POI corpus construction method and used Word2Vec model to acquire the distributed semantic representation of the POI. An evaluation metric was developed to measure whether our proposed method could effectively capture the environmental semantics and type characteristics of the POI, and also prove the association between POI spatial distribution and their types.
- The grid division was constructed to realize the distributed representation of the regional feature information of the candidate ROI. After the calculation method of the similarity score between the ROI vector and query condition was designed to achieve the top-K query of the keyword-based ROI, we verified that the proposed method achieved significant improvements over the baselines in a large-scale real dataset.
- Considering the extension of the ROI multi-keyword query mode, we demonstrated the validity and feasibility of the multi-keyword query.

The rest of this paper is organized as follows. Section 2 overviews the related work. Section 3 formally defines the statement of our problem. Our method is elaborated in Section 4. Section 5 proves the efficiency of the POI vectors and evaluates the performance of the proposed method by comparison. Finally, we briefly conclude the paper in Section 6.
2. Related Works

Spatial keyword search, which is a fully developed research field in geographic information retrieval, has achieved valuable and significant research results [10,15–17]. The related research methods can be mainly divided into spatial-first approaches and keyword-first approaches [18]. While the spatial-first approaches usually perform geographic queries on spatial indexes to obtain the target objects [19], the keyword-first approaches are more relevant to our work, where existing research has focused on the textual association between the target geographic object and the query description to find relevant locations (such as POI) [20,21]. Unlike the traditional POI-based query methods above-mentioned, our study emphasized the ROI exploration related to the spatial keyword query. There have been some studies on ROI search in recent years, which can be roughly divided into two different research methods: a search based on density and a search based on fixed division. The former attempts to explore the density of the POIs related to the query keywords in the research region and obtain the POI cluster as the target ROI by a density-based clustering algorithm to realize the relevant ROI exploration [6–8]. However, as the scale of POI datasets increases, this approach usually results in excessive time consumption, and it is quite difficult for this approach to control the size of the ROIs.

The approach based on fixed partition can compute the correlation between each fixed region and the query, which normally adopts the grid index to construct the region division [22,23]. One of the main challenges in this approach is that there are some difficulties in measuring the similarities between the query and each ROI. To solve this problem, Fan J. et al. [9] proposed an ROI exploration method that utilized the ROI as a query description to search the most relevant regions by comparing the spatial overlap and textual similarity between the candidate regions and the query ROI. However, in our work, the query keywords were considered as the input of the query, which is more convenient for users to understand and operate. Similarly, using textual keywords, Zhi Y et al. [11] proposed a method based on the density statistics of POIs with related keywords in the region to measure the correlation between the target region and the query, whereas it ignored abundant and available POI environmental information. Compared with the above methods, the biggest difference of this paper is the introduction of the concept of the environmental semantics of spatial objects. With the distribution features and semantic expressions of each spatial object captured by the statistical model based on deep learning, the regional features corresponding to the candidate ROI were constructed to match the query keywords.

The spatial semantic representation of spatial objects, such as POI, refers to the semantic distributed representation of the word embedding technique in natural language processing (NLP), which was first proposed in [24]. The related research on the language model can be traced back to [25], which mentions the feasibility of the learning language model via a neural network. One of the most classic works related to the language model is [26], which laid the foundation for the language model and word embedding technique. With the rapid development of related technologies in deep learning, the Google Word2Vec model has been proposed and improved [27,28] and is considered to be one of the most successful deep learning language models. Its training models mainly include Skip-Gram and Continuous-Bag-of-Words, and the optimization process can be divided into Hierarchical Softmax [29] and Negative Sampling [30], which can speed up the training steps and convergence. Relevant research in recent years has fully proven that the WORD2VEC model can effectively capture the similarity between words by running it on a large-scale corpus. It is widely used in the pretreatment of machine translation, interest recommendation, information retrieval, and other fields, and has obtained remarkable results.

Recent years have witnessed some related applications, such as the use of geographic information for urban computing and pattern discovery by the embedding technique. Some of these applications consider running embedding operations on geo-tagged spatial text data to explore spatial information. Based on geo-tagged tweets, [31] they attempted to adopt word embedding to explore the impact of the geographical distribution on the semantics of the spatial texts, but their work focused on linguistics and topic discovery. The deep representation of trajectory information was learned from
Association for Information Systems (AIS) data and recognized the clustering and analysis on trajectory characteristics [32]. Mai G. et al. [33] adopted Doc2vec to transform the description of each historic place from DBpedia into a paragraph vector and performed a clustering algorithm to implement semantically enriched geospatial data visualization. Nevertheless, the above-mentioned studies did not directly study the distribution characteristics of spatial objects because the main operation object was still text data. In contrast, we directly captured the geographical environmental characteristics of the POIs by constructing spatial contexts and exploring the deep dependencies between the geographic distribution and spatial object types. The closest work to ours was conducted in [14], which designed a greedy algorithm to directly model the spatial objects by POI embedding and used the acquired POI vectors for the downstream urban land classification task. In contrast, we constructed a more realistic and natural POI corpus for the model training and focused on exploring the spatial characteristics and environmental semantics of the ROIs to implement a spatial keyword-based ROI query. Other similar works have studied POI recommendations by using POI embedding to provide high-quality data input in the pretreatment of the prediction model. The POI embedding was obtained to predict the next visitor of a POI point by incorporating spatial information into the Word2Vec model [34].

However, our work did not take time sequential data as the research object, but considered exploring the relevant characteristics of spatial objects and understanding their environmental semantics to construct the distributed representation of POI, so that a similarity match between the ROI spatial description and query could be realized [35].

3. Problem Statement

Given a raw POI dataset $P$ in a limited map, each point is assigned with an exact coordinate location $(x, y)$. $x$ and $y$ are the longitude and latitude, and it also has a type label, $t_i$. Other unnecessary attributes (e.g., Name, Address, and Alias) were ignored in this paper. Table 1 gives an example of the POI data. Based on the POI dataset, the intention of ROI exploration was to find a close related region populated with various POIs for the users’ keyword-based query.

Table 1. Example of raw dataset $P$.

<table>
<thead>
<tr>
<th>ID</th>
<th>Location</th>
<th>Type Label</th>
<th>Other Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(116.30, 40.41)</td>
<td>library</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>(116.43, 39.95)</td>
<td>newsstand</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>(116.46, 39.96)</td>
<td>Starbucks</td>
<td>...</td>
</tr>
<tr>
<td>4</td>
<td>(116.41, 39.98)</td>
<td>clinic</td>
<td>...</td>
</tr>
</tbody>
</table>

First, the ROI keyword query was defined as follows:

**Definition 1.** ROI keyword query: Given a keyword set $Q = \{q_1, q_2, q_3, \ldots, q_n\}$, each $q_i$ can be mapped to a certain type label $t_i$. The query $Q$ expects to find some regions whose characteristics are the most relevant to these specific requests $q_i$.

Second, the conception of ROI was specifically described as follows in this paper:

**Definition 2.** ROI: ROI is a relevant region $R$ where a certain number of POIs satisfy the query location. With a POI regarded as an atom in this region, $R$ is represented as a mixed POI set $R = \{p_1, p_2, p_3, \ldots, p_n\}$, where $p_i$ is a POI with one type label, $t_i$. After the ROI division, each region is viewed as the candidate ROI to be matched. More details will be explored in-depth in Section 4. For now, the ROI can be treated as an abstract set.

**Definition 3.** Top-K similarity search: By dividing the raw POI data into $n$ candidate ROIs, the similarity score between each one and the query $Q$ can be calculated. Then, the search will return a sorted top-K collection of the ROIs with the $K$ highest similarity score.
An instance of the top-K similarity search is shown in Figure 1. The query \( Q \) group is \{school\}. There were four candidate ROIs to be matched. Assuming \( K = 1 \), the ROI colored with red was returned as the top-1 result. It is worth mentioning that the similarity calculation takes into account the environmental semantics of the regions. On the basis of the well-trained distributed representation of the POIs, our method will generate the corresponding vector for each candidate ROI, which contains the internal environmental information and structural characteristics of the ROI, i.e., the environmental semantics of the region. Thus, the vector corresponding to the query keyword will be treated as the search condition to find the top-K ROIs that match the query vector. The example shown in Figure 1 calculates the similarity score between the vector corresponding to the query keyword \( Q \) [school] and each candidate ROI vector to find the top-1 result.

![Figure 1](image_url)

**Figure 1.** The blue points represent the buildings with type label “school” and the yellow points indicate the buildings with type label “residential buildings”. As a result, the red region is returned as the result of our top-1 query by matching the environmental information of each candidate Region-Of-Interest (ROI) for the query.

The symbols used in this paper are summarized in Table 2.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P )</td>
<td>a collection of POI</td>
</tr>
<tr>
<td>( p_i )</td>
<td>i-th POI</td>
</tr>
<tr>
<td>( t_i )</td>
<td>i-th type label</td>
</tr>
<tr>
<td>( Q )</td>
<td>a keyword query group</td>
</tr>
<tr>
<td>( q_i )</td>
<td>i-th keyword in query</td>
</tr>
<tr>
<td>( R )</td>
<td>an ROI</td>
</tr>
</tbody>
</table>

### 4. Methods

#### 4.1. The Overall Architecture

The workflow diagram of our method can be seen in Figure 2. First, we describe the data that were used to train our POI vectors in Section 4.2, which also considered the data as the input of the whole workflow. According to our specific intentions, the procedure of the workflow was composed of three steps:

First, the raw data, containing a large number of POIs with type labels, were used to construct the corpus (an organized computer-readable collection of text or speech in the field of NLP) of the POIs. The skip-Gram model of Word2Vec was trained over the POI corpus to express POIs in a high-dimensional space, which could capture their semantic information and environmental state. The latent semantic association of POI embedding vectors is revealed in the correlation analysis (Section 4.3).

Second, a grid division in the research region was built to acquire the candidate ROIs, each of which was viewed as a POI set. The candidate ROIs could be described as vectors by the product of
Step 1 (POI embedding vectors). At the same time, two variant methods of generating candidate ROIs were introduced to make the ROI vector description more reasonable (Section 4.4).

Finally, the products of the previous step, the candidate ROI vectors, were considered as the inputs to this step. They were utilized to calculate the relevance score by the similarity formula with the user’s keyword query group Q. Therefore, based on different query modes, the top-K ROIs related to the user’s query are returned as the final results (Section 4.5).

In the remainder of this section, we present further details on the specific process of these steps.

![Figure 2. Workflow of the spatial keyword query of the ROI with the distributed representation of Point-Of-Interest (POIs). TF-IDF, term frequency-inverse document frequency.](image)

### 4.2. Data Description

In this paper, 379,790 records of Beijing POIs with multi-level type labels were fetched via the Application Programming Interfaces (APIs) of the Amap Service [36], which is one of the most popular map services in China. A type label is made up of three levels, where the lower category is attached to the higher category. A lower category level usually means that there are more detailed descriptions and more specific restrictions about the POI. For example, given a POI type labeled “Science and Education Service–School–university”, its top-level is “Science and Education Service”, the middle-level is “School”, and the bottom-level is “university”. Moreover, there are similarities among the POI types of the same middle-level type or same top-level type. It is noted that “Science and Education Service–School–university” is similar to “Science and Education Service–School–Middle School” because both of them belong to the middle-level “School” and the top-level “Science and Education Service”. We kept the bottom-level types that appeared more than 10 times in the entire dataset, which were viewed as the words in our training model. As a result, there were 19 top-level types, 174 middle-level types, and 521 bottom-level types in our POI dataset. The type and count of each top-level POI category is shown in Table 3. Each top-level type was designed with an ID for ease of description in the following analysis. Bottom-level types were considered as type labels to construct the POI corpus.
Table 3. Type and count of top-level POI categories.

<table>
<thead>
<tr>
<th>Top-level Type</th>
<th>ID</th>
<th>Count</th>
<th>Top-level Type</th>
<th>ID</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shopping Service</td>
<td>1</td>
<td>76,038</td>
<td>Financial Insurance Service</td>
<td>11</td>
<td>11,503</td>
</tr>
<tr>
<td>Life Service</td>
<td>2</td>
<td>57,178</td>
<td>Car Service</td>
<td>12</td>
<td>10,866</td>
</tr>
<tr>
<td>Transportation Facility</td>
<td>3</td>
<td>37,404</td>
<td>Accommodation Service</td>
<td>13</td>
<td>7487</td>
</tr>
<tr>
<td>Catering Services</td>
<td>4</td>
<td>36,001</td>
<td>Public Facility</td>
<td>14</td>
<td>5376</td>
</tr>
<tr>
<td>Government Agency</td>
<td>5</td>
<td>30,484</td>
<td>Car Maintenance Station</td>
<td>15</td>
<td>2196</td>
</tr>
<tr>
<td>Science and Education Service</td>
<td>6</td>
<td>26,726</td>
<td>Road Auxiliary Facilities</td>
<td>16</td>
<td>2189</td>
</tr>
<tr>
<td>Company</td>
<td>7</td>
<td>21,011</td>
<td>Famous Tourist Sites</td>
<td>17</td>
<td>2090</td>
</tr>
<tr>
<td>Residence</td>
<td>8</td>
<td>19,137</td>
<td>Car Sales</td>
<td>18</td>
<td>649</td>
</tr>
<tr>
<td>Healthcare Service</td>
<td>9</td>
<td>16,893</td>
<td>Motorcycle Service</td>
<td>19</td>
<td>352</td>
</tr>
<tr>
<td>Sports and Leisure Service</td>
<td>10</td>
<td>16,210</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3. POI Embedding

As mentioned in the introduction, solely counting the number of the POIs with labels to match the ROI will result in neglecting their spatial distributions and environmental information. To solve this problem, some recent works in NLP have inspired us, as the distributed representation model Word2Vec can capture the semantic relations in each word’s context and produce a high-quality collection of word embedding vectors encoding latent semantic information [27,28]. In addition, the distribution of the POI group size in [37] revealed that the type frequency of these POIs conformed to a power distribution, which is similar to the word frequency distribution in documents [38]. This means that the same approach can be used to capture the environmental semantics of POIs, which is verified explicitly in [14]. Each POI with a type label (bottom-level type) is transformed into a high-dimensional vector, so this step was named POI embedding.

4.3.1. Corpus Construction

To obtain a meaningful POI embedding vector, an organized POI corpus was prepared before the training step of the raw data. It can be seen that there is an obvious difference between the word corpus and the POI corpus. Compared with the POI corpus, the word corpus consists of many ordered documents with words in a natural sequential order. Thus, it is necessary for the POI corpus to be reorganized in a new way that is similar to the word order. The key to this problem is to define the spatial context of a certain POI and provide a reasonable input for the Word2Vec model. To sufficiently capture the spatial distribution and type correlation of a certain POI, we iterated every POI in the raw data and found its corresponding spatial context. The type label of the center POI is denoted as $t_i$. Its spatial context is denoted as a set $T_{context} = \{t_{i-nc}, \ldots, t_{i-1}, t_{i}, t_{i+1}, \ldots, t_{i+nc}\}$, which are the type labels of the $2c$ nearest POI neighbors to the center POI in the coordinate system. For every $T_{context}$, we can obtain $2c$ Cartesian products $(t_j, t_k)$ as the training pairs for each center POI $t_i$, which is similar to the sliding window in the Word2Vec model. The exact coordinate location of each POI was given in the raw dataset so that we could successfully obtain the spatial context of each POI to build our POI embedding corpus. Furthermore, to accelerate the construction of the training data, we iterated all POIs and found the $2c$ nearest neighbors of each POI by spatial indexing techniques, such as R-tree and Geohash. Compared with the TAZ-POI corpus in [14], the corpus constructed by our method is more natural and convincing in catching the inner space relationships and exploring the correlations of POI types.

4.3.2. Training POI Vectors by the Skip-Gram Model

With these training sets fed to Word2Vec, the Skip-Gram model of Word2Vec was adopted to achieve POI embedding. The basic framework of the Skip-Gram model is shown in Figure 3, which attempts to use the center POI type to predict its spatial context POIs and learn all of the word embedding vectors.
Based on the neural network language model (NNLM) and Naive Bayes model, assuming the generation of each \( t_x \) is independent, the context type probability distribution learned from this model is defined as

\[
y'(t_{i-1}, t_{i-1}, t_{i+1}, \ldots, t_{i+c}) = \prod_{i-c \leq j \leq i+c} p(t_j | t_i).
\]  

(1)

In Equation (1), \( p(t_j | t_i) \) is the normalized conditional probability of predicting a certain context POI type \( t_x \) from the center type \( t_i \). \( y' \) is the joint probability distribution of the context labels that the model can learn from the training data. The original likelihood distribution \( y \) of the context labels follows a multinomial distribution. To conform \( y' \) to the true probability distribution of the POI types \( y \) in the raw data, a cross entropy is used as the loss function, which measures the gap between the two probability distributions as follows:

\[
f(\theta) = -y \log(y') = -\sum_{i-c \leq j \leq i+c} \log p(t_j | t_i).
\]  

(2)

In Equation (2), minimizing the loss for a center POI \( t_i \) can be utilized to optimize the learning process and adjust the weight matrix in the hidden layer. The essence of this model is to calculate the similarity between the vector of input word \( t_i \) and the vector of its context word \( t_x \) and then perform a softmax normalization. Therefore, when a one-hot vector is regarded as the input vector, the vector of its context words will be in the form of a softmax representation, which reveals that the context vector should belong to a certain type. This procedure, called forward computation in deep learning, further describes the conditional probability \( p(t_x | t_i) \) as

\[
p(t_x | t_i) = \frac{\exp(V_i^T \cdot V_x)}{\sum_{j=1}^{K} \exp(V_j^T \cdot V_x)}.
\]  

(3)
In Equation (3), $V$ is the $D \times K$ weight matrix where $V_t$ is the column vector corresponding to the center POI type $t_i$, and $V_x$ is the column vector corresponding to the context POI type $t_x$. $D$ is our presupposed dimension in the distributed representation of POI types, while $K$ is the total number of our POI types. The softmax function is leveraged to calculate the conditional probability of the type $t_x$ in $K$ types for the center type $t_i$.

Combined with the above explanation, the loss function is redefined as follows:

$$L(\theta) = -\frac{1}{T} \sum_{i=1}^{T} \left( \sum_{i < s < i + c, s \neq i} \log \frac{\exp(V_i^T \cdot V_s)}{\sum_{j=1}^{K} \exp(V_j^T \cdot V_s)} \right)$$

(4)

where $T$ is the amount of our POI corpus. It was noticed that the optimization to minimize the loss for all of the data is a time-consuming step. To accelerate this process, two optimization algorithms, mini-batch gradient descent and negative sampling, were implemented in this model [30].

After training, an adjusted weight matrix $V$, which can make the learned distribution consistent with the true one, is returned. For a certain type $t_j$, we can look up the POI embedding vector $v_j$ from the $j$-th column vector in $V$. As the Word2Vec model considers the impact of the environment around each POI, the POI vector corresponding to a type will be reflected in its environmental semantics. These POI embedding vectors capturing the environment information of each POI type are considered as the input for the next stage.

4.3.3. Correlation Analysis of the POI Vectors

We utilized these POI vectors for clustering and correlation analysis to reveal that the POI vector could reflect the type association effectively. The similarity score between the POI vectors can be calculated by cosine similarity [39]. Next, the k-means++ [40], which improves the initialization of k-means clustering and reduces the error, was implemented to cluster the POI vectors and quantify the relevance between the POI types. However, how to decide the number of clusters $K$ is still remains a question. To cope with this issue, the average silhouette coefficient [41] based on cosine similarity can be used to evaluate the effect of the POI vector clustering and determine an appropriate cluster number $K$. The silhouette coefficient of the POI $p_i$ is denoted by $s(i)$, whose range is $[-1, 1]$. A value of $s(i)$ close to 1 shows that the POI is clustered effectively and far from other clusters; a value of $s(i)$ close to 0 means that there is some difficulty in judging the belonging of the POI; a value of $s(i)$ close to $-1$ usually means it is put into the wrong cluster. As a whole, the average silhouette coefficient (ASC) of the entire dataset can reflect the appropriateness of clustering. In general, a larger ASC means a more reasonable result for the cluster number $K$. More implementation details of the association analysis will be demonstrated in the Experimental and Results Section.

4.4. Candidate ROI Vector Generation

After obtaining the well-trained POI embedding vectors, an ROI regarded as an abstract set of POIs can be obtained by using the POI vectors included in it. Therefore, candidate ROI vector generation becomes the key issue. It is necessary to select a reasonable division for candidate ROI and generate ROI vectors based on internal POI embedding vectors. It is worth mentioning that in most daily scenarios, users focus more on the location of regions strongly related with the query, rather than the specific shape of the regions. Consequently, we referred to the grid-based approaches [11,22,23] and established our region grid division. Compared to other methods that can explore the shape of the regions by the density of points such as DBSCAN (its time complexity is $O(n^2)$, $n$ is the number of POI), the grid division can effectively improve the construction of the candidate ROIs (its time complexity is $O(n)$). Meanwhile, it can also return acceptable results, meeting the user’s query requirements by setting different scales of the grid size.
4.4.1. Grid Division

The research region in the geographic coordinate system, where many POIs with labels are located, is converted into a rectangular region by the projection transform. The region’s length transformed by longitude span is \( l \) km and its width transformed by latitude span is \( w \) km. Next, we divided it into \( a \times b \) grids with a length of \( l/a \) km and width of \( w/b \) km, where \( a \) and \( b \) are the parameters determined by the user to control the area of the grid. All POIs are put into their corresponding grids based on their coordinate positions.

Then, each grid considered as the candidate ROI contains a certain number of POIs with labels (it is noted that there is no POI in some grids). Given the POI vectors, the ROI vector can be computed by aggregating the POI vectors in it. An intuitive method is to calculate the weighted mean of the POI vectors in candidate ROIs by their frequency of occurrence:

\[
R_i = \frac{\sum_{j=1}^{K} w_j v_{tj}}{\sum_{j=1}^{K} w_j}
\]

where \( R_i \) and \( v_{tj} \) represent the \( i \)-th ROI vector and the POI vector with type label \( t_j \), respectively. The weight \( w_j \) is the frequency of the POI with label \( t_j \) in \( R_i \).

4.4.2. TF-IDF Method

However, considering that some infrequent POIs tend to have a negative impact on the function and type of the ROI, term frequency-inverse document frequency (TF-IDF), a common method in information retrieval that can evaluate the importance of words in a corpus, was utilized to adjust the weights of POI types in each ROI. The principle is that the importance of one word increases proportionally with its appearance in a document, but decreases inversely with its frequency in the whole corpus.

It inspired us that all of the candidate ROIs can be viewed as documents, while each POI with type label can be viewed as a word. Thereby, the IDF of the POI with type label \( t_j \) is

\[
IDF(t_j) = \log \left( \frac{N + 1}{N(t_j) + 1} \right) + 1.
\]

In Equation (6), \( N \) is the total number of the candidate ROIs and \( N(t_j) \) is the number of the ROIs including the POI with \( t_j \). The constant term is to avoid a zero denominator. IDF reflects the frequency of the POI with \( t_j \) in all of the candidate ROIs. A high value of the IDF indicates that the POI with \( t_j \) appears in most candidates. In contrast, the low IDF means it is rare in the whole POI corpus. Therefore, the TF-IDF, the weights of the POI vector with type \( t_j \), can be recalculated as follows:

\[
w_{ij} = TF(t_{ij}) \times IDF(t_j)
\]

where \( TF(t_{ij}) \) is the frequency of the POI with label \( t_j \) in the \( i \)-th ROI. The formula of the ROI vector is updated by new weights. By leveraging the TF-IDF method, we can construct a more reasonable candidate ROI vector, making its regional characteristic expression more consistent with the realistic environment.

Algorithm 1 describes the detailed implementation of the TF-IDF method, which is processed after grid division. Each \( S_i \) in the candidate ROI set \( S \) is a POI set, where each POI has a type label \( t_j \) corresponding to POI vector \( v_{tj} \). First, the inverse document frequency of each \( t_j \) is calculated, and then the corresponding weights \( w_j \) of each POI vector are calculated for each \( S_i \) (lines 5–6). Each candidate ROI vector \( R_i \) can be obtained by Equation (5), where the POI vector set \( v \) will be utilized. Eventually, it returns the candidate ROI vector set \( R \).
Algorithm 1: TF-IDF Method

Input: (1) candidate ROI set $S$ (2) POI vectors set $v$ (3) type labels set $t$

Output: candidate ROI vectors set $R$

1: for each $t_j \in t$ do
2: $IDF(t_j) =$ result by Equation (6)
3: for each $S_i \in S$ do
4: for each $t_j \in t$ do
5: $TF(t_j) =$ the frequency of POI with label $t_j$ in $i$-th ROI
6: $w_j = IDF(t_j) \times TF(t_j)$
7: $R_i =$ result by Equation (5)
8: return $R$

4.4.3. Gaussian Kernel

On the other hand, it is noteworthy that there is an external association between a certain ROI and its surrounding regions in geographic space. With the relevance involved, the result will be more robust and closer to the realistic situation. In order to improve the quality of the ROI vector, we represent the center ROI vector as the weighted mean of its surrounding ROIs and itself. According to the principle that the correlation decays inversely with the increase of distance [42], the relevance between the center ROI and its surrounding ROI can be assumed to obey the two-dimensional Gaussian distribution in geographic space [11]. Then, we introduced the Gaussian kernel and calculated the weighted average, which is similar to the convolution operation of the image. The center ROI vector is adjusted as follows:

$$R'(i, j) = \frac{\sum_{m=0} \sum_{n=0} K(m,n) \ast R(i - m, j - n)}{\sum_{m=0} \sum_{n=0} K(m,n)}.$$  \hspace{1cm} (8)

In Equation (8), $R$ is the original ROI matrix and $R(i-m, j-n)$ is the ROI vector in the position $(i-m,j-n)$. Similarly, $R'$ is the adjusted ROI matrix and $R'(i,j)$ is the updated vector for the center ROI $R(i,j)$. $K$ is the Gaussian kernel where $K(m,n)$ represents the weight of these ROI vectors that are involved for calculating the center ROI $R(i,j)$. With a $3 \times 3$ Gaussian kernel taken as an example, the specific process is shown in Figure 4.

Using the Gaussian kernel computing center ROI, we can effectively take into account the impact of the surrounding ROIs on the central ROI. In order to reduce the computational complexity, we only took one-hop adjacent ROIs to the center into account and adopted a $3 \times 3$ Gaussian kernel in this paper.

Considering the parameters ($a$ and $b$) of the grid division, candidate ROI vectors set $R$ can be represented as the vectors matrix $R(a,b)$ in Algorithm 2. First, lines 1–2 perform the expansion and filling process shown in Figure 4. Next, the convolution multiplication of Equation (8) is performed for each unexpanded ROI vector $R(i,j)$ on the augmented matrix. As a result, an adjusted candidate ROI vectors matrix $R'(a,b)$ will be returned.

Algorithm 2: Gaussian Kernel Method

Input: (1) candidate ROI vectors matrix $R(a,b)$ (2) convolution kernel $K$

Output: adjusted candidate ROI vectors matrix $R'(a,b)$

1: expand $R(a,b)$ to the size of $R(a+2, b+2)$
2: fill the expended parts with 0 vector
3: for each $R(i,j) \in R(a+2, b+2)$ do
4: if $R(i,j) \notin$ the expended parts then
5: $R'(i,j) =$ result by Equation (8)
6: return $R'(a,b)$
The POI vectors imply the environmental semantics and distribution information of the POIs with various type labels, and the POIs with the same spatial distribution tend to have similar category regional characteristics, which means that the correlation between the different type of labels can be measured by the similarity score between the POI vectors. In general, the cosine similarity is considered to be one of the most appropriate methods for calculating the similarity between the vectors in high-dimensional space. Therefore, the formula of the similarity score between POI vectors corresponding to two different types $t_a$ and $t_b$ is

$$\text{Similarity}(v_{t_a}, v_{t_b}) = \frac{v_{t_a} \cdot v_{t_b}}{|v_{t_a}| |v_{t_b}|}.$$  \hspace{1cm} (9)

The similarity score is in the range of $[-1, 1]$. A greater similarity score indicates that there is a strong correlation between their corresponding types. This point is further demonstrated in Section 5.

Similarly, given the ROI vector for each candidate ROI and the single keyword $q_{tx}$, which is mapped to the POI vector $v_{tx}$ with label $t_x$, the similarity score between the query and each candidate ROI can be calculated by

$$\text{Similarity}(q_{tx}, R_i) = \frac{v_{tx} \cdot R_i}{|v_{tx}| |R_i|}$$  \hspace{1cm} (10)

where $R_i$ represents the $i$-th candidate ROI to be matched. The ROI vector comprehensively considers its own composition of the POIs and the impact from the surrounding environment to express the regional characteristics of each candidate ROI intensively. Compared with the simple statistics of these

**Figure 4.** Gaussian kernel computing. The figure reveals that the first step of computing is to expand the original ROI vector matrix $R(a,b)$ to the size of $R(a+2, b+2)$, where the extended part is filled by the zero vector. Meanwhile, with the convolution kernel weight corresponding to the $\theta$ vector region set as zeros, the ROI vector in the edge of the original matrix can also be computed.

### 4.5. Query Search

In this subsection, we define the query modes as the single-keyword ROI query and the multi-keyword ROI query, respectively, according to the number of query keywords, and propose a method to measure the similarity between keyword query and candidate ROIs.

#### 4.5.1. Single-Keyword Query Mode

The POI vectors imply the environmental semantics and distribution information of the POIs with various type labels, and the POIs with the same spatial distribution tend to have similar category characteristics, which means that the correlation between the different type of labels can be measured by the similarity score between the POI vectors. In general, the cosine similarity is considered to be one of the most appropriate methods for calculating the similarity between the vectors in high-dimensional space. Therefore, the formula of the similarity score between POI vectors corresponding to two different types $t_a$ and $t_b$ is

$$\text{Similarity}(v_{t_a}, v_{t_b}) = \frac{v_{t_a} \cdot v_{t_b}}{|v_{t_a}| |v_{t_b}|}.$$  \hspace{1cm} (9)

The similarity score is in the range of $[-1, 1]$. A greater similarity score indicates that there is a strong correlation between their corresponding types. This point is further demonstrated in Section 5.

Similarly, given the ROI vector for each candidate ROI and the single keyword $q_{tx}$, which is mapped to the POI vector $v_{tx}$ with label $t_x$, the similarity score between the query and each candidate ROI can be calculated by

$$\text{Similarity}(q_{tx}, R_i) = \frac{v_{tx} \cdot R_i}{|v_{tx}| |R_i|}$$  \hspace{1cm} (10)

where $R_i$ represents the $i$-th candidate ROI to be matched. The ROI vector comprehensively considers its own composition of the POIs and the impact from the surrounding environment to express the regional characteristics of each candidate ROI intensively. Compared with the simple statistics of these
POIs with the keyword, the impacts of each POI point on the regional characteristics were all taken into consideration. If there is a high similarity score between the ROI vectors indicating the overall features of this region and the vectorized query keyword, it is reasonable to think that the ROI area is closely related to this query. After obtaining the similarity scores for all candidate ROIs, we performed a sort operation to find the top-K results with the highest scores. As the sorting process was not our research focus, we just implemented a simple bubble sort algorithm.

4.5.2. Multi-Keyword Query Mode

For the multi-keyword query group \( Q = \{q_1, q_2, q_3, \ldots, q_n\} \), the mean of the vectors corresponding to all of the keywords is calculated as the final query vector \( V_Q \), thus its similarity with each candidate ROI is measured by Equation (10). The rationality of the design is shown in the ROI, whose characteristics meets the environment of all keyword POIs, if it can highly match the multi-keyword query. If the ROI lacks the element of the POI corresponding to a certain keyword, its vector will have a large angle with the query vector in the high-dimensional vector spaces, so the cosine similarity is relatively low and does not rank high in the result of the multi-keyword query.

As there is a high similarity between the two query modes, the query search implementation of them will be shown in Algorithm 3 together. Lines 1–5 generate the query vector group \( Q_v \) corresponding to the keyword query group \( Q \). Lines 6–7 perform the average operation on \( Q_v \). At this time, regardless of whether it is a single keyword query or multi-keyword query, the output is the average query vector \( Q_{\text{mean}} \). Then, the similarity score between each candidate ROI vector \( R_i \) and \( Q_{\text{mean}} \) is calculated by Equation (10), which can be used to sort the candidate ROI vector \( R_i \) in descending order. Finally, it returns the top-K ROI \( R_{\text{top-K}} \) relevant to the query.

<table>
<thead>
<tr>
<th>Algorithm 3: Query Search</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> (1) candidate ROI vectors set ( R ) (2) keyword query group ( Q ) (3) parameter ( K ) (4) POI vectors set ( v ) (5) type labels set ( t )</td>
</tr>
<tr>
<td><strong>Output:</strong> The top-K ROIs related to query ( R_{\text{top-K}} )</td>
</tr>
<tr>
<td>1: ( Q_v = {\emptyset} )</td>
</tr>
<tr>
<td>2: for each ( q_j \in Q ) do</td>
</tr>
<tr>
<td>3: for each ( t_m \in t ) do</td>
</tr>
<tr>
<td>4: if ( q_j = t_m ) then</td>
</tr>
<tr>
<td>5: append ( v_tj ) into ( Q_v )</td>
</tr>
<tr>
<td>6: if ( Q_v \neq {\emptyset} ) then</td>
</tr>
<tr>
<td>7: ( Q_{\text{mean}} = \text{mean}(Q_v) )</td>
</tr>
<tr>
<td>8: for each ( R_i \in R ) do</td>
</tr>
<tr>
<td>9: ( \text{Similarity_Score}(R_i) = \text{Similarity}(Q_{\text{mean}}, R_i) ) by Equation (10)</td>
</tr>
<tr>
<td>10: sort ( R ) in descending order of ( \text{Similarity_Score}(R) )</td>
</tr>
<tr>
<td>11: return top-K ( R_{\text{top-K}} ) in ( R )</td>
</tr>
</tbody>
</table>

5. Experiment and Results

In this section, our work and experimental results were evaluated in two steps. In the first step, we proposed an evaluation schema to study how to define the training parameters of our model and obtained the POI vectors in high quality. Then, we clustered these vectors to conduct a correlation analysis, verifying the effectiveness of the POIs for environmental semantic expression. In the second step, we compared our ROI query method based on these POI vectors and its variants with the baseline method to verify their effectiveness in ROI exploration. Finally, we present the results in a real dataset to reveal the feasibility of the proposed method in a multi-keyword query.
5.1. POI Vector Acquisition and Analysis

5.1.1. Training POI Vectors and Parameter Selection

We trained the POI embedding vectors by utilizing the Word2Vec model in TensorFlow 1.11.0. [43]. Specifically, the iterations were set to 20 and the parameters were set to the default values except the window size \( c \) and the dimension of the embedding vector \( D \). It was noted that the two important parameters for Word2Vec, \( c \) and \( D \), directly determined the quality of the acquired POI vectors. As the number of the POI types is much smaller than the size of vocabulary in the real world, we selected \( c \) from 1 to 20 with a step interval of 1 and \( D \) from 10 to 200 with a step interval of 10.

Evaluation metric: In the Introduction, we mentioned that there was relevance between type association and geographic distribution. We also illustrated that our method could effectively capture the environmental characteristic and geographic distribution of the POIs by the Word2Vec model. It can be inferred that whether POI vectors can reflect the original type association is considered as the evaluation standard for the quality of the POI vectors. Therefore, we designed a rule based on the original multi-level types to evaluate the similarity score between two POI types:

- If they have the same bottom-level type, the similarity score is 1;
- If they have the same middle-level type, but different bottom-level type, the similarity score is 0.5;
- If they only have the same top-level type, the similarity score is 0.25;
- If they have nothing in common for multi-level types, the similarity score is 0.125.

Meanwhile, we can obtain the similarity score between the POI vectors from our parameter iterations as mentioned in Equation (10). Next, suppose that the similarity scores between the POI vectors generated by the iteration of window size \( c \) and vector dimension \( D \) is \( X_{c,D} \), which is a \( 521 \times 521 \) similarity score matrix. The corresponding score from the original multi-level types is \( Y \), which is also a \( 521 \times 521 \) similarity score matrix based on the designed rules. Then, the correlation between the two variables is given by the Pearson correlation coefficient:

\[
rc,D = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2 \sum_{i=1}^{n}(Y_i - \bar{Y})^2}}. \tag{11}
\]

The Pearson correlation coefficient between variables \( X_{c,D} \) and \( Y \) is defined as the quotient of the covariance and standard deviation between them. The absolute value of the correlation coefficient \( |r_{c,D}| \) reveals the strength of the correlation: the closer the correlation coefficient is to 1 or \(-1\), the stronger the correlation; the closer the correlation coefficient is to 0, the weaker the correlation. In our metric, a larger positive Pearson correlation coefficient \( r_{c,D} \) indicates that the POI vector of the iteration \((c,D)\) is more in line with the original multi-level type association, which also means it is of high quality at this time.

Figure 5 shows the change in Pearson correlation coefficient with different window size and vector dimension using our evaluation method. It can be seen that as the window size and dimension continue to increase, the correlation coefficient change gradually slows down and tends to be stable at the end. We obtain the value \((c = 14, D = 120)\) in the platform region that maximizes the correlation coefficient, i.e., \( r_{max} = 0.32784 \). It can be considered that the parameters at this iteration produce results with the best quality and the POI vectors will be used for the experiments below.
5.1.2. Correlation Analysis

After parameter selection of the training model, the high-quality POI embedding vectors were utilized for correlation analysis to reveal their potential semantic relevance according to the evaluation presented in Section 4.3.3. Figure 6 shows the clustering results with different cluster number $K$. When $K = 2, 5, 7$, it can be seen that there was a maximized average silhouette coefficient (ASC). On the other hand, the error square sum (SSE) for the POI vectors was also taken into account as evaluation criteria. As an exorbitant SSE is usually considered bad performance, when $K = 2$, the number was not adopted as a valid result.

**Figure 5.** Parameter selection of the distributed representation of POIs. The X-axis is the window size, the Y-axis is the dimension, and the z-axis is the Pearson correlation coefficient corresponding to the first two. The different colors indicate the magnitude of the Pearson correlation coefficient.

**Figure 6.** Change of average silhouette value (left y-axis) and error square sum (right y-axis) of clustering results (POI vectors) with increases of $K$ value (x-axis).
The proportion of the different top-level types is calculated by the bottom-level types in each cluster in Table 4, which reveals that the clustering results are meaningful.

Table 4. Clustering results. A higher percentage value means that the cluster has a higher proportion in top-level types, that is, it is more similar to this type. The numbers denote the type IDs of the top-level types, for example, “1” represents the “Shopping Service”.

| K | Cluster 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 2 | C1      | 0.0% | 2.1% | 1.1% | 0.0% | 3.2% | 0.0% | 0.0% | 0.0% | 0.0% | 2.1% | 10.5% | 0.0% | 0.0% | 36.6% | 3.2% | 0.0% | 41.1% | 0.0% |
| 5 | C2      | 73.1% | 7.1% | 4.8% | 9.1% | 6.1% | 5.1% | 5.1% | 2.0% | 1.0% | 4.0% | 3.6% | 0.0% | 2.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 3 | C3      | 0.0% | 6.8% | 6.8% | 0.0% | 0.0% | 8.5% | 11.9% | 3.4% | 0.0% | 89.0% | 0.0% | 0.0% | 1.7% | 0.0% | 0.0% | 0.0% | 0.0% | 22.0% | 0.0% | 0.0% |
| 4 | C4      | 49.5% | 2.3% | 15.1% | 31.4% | 0.0% | 1.2% | 0.0% | 0.0% | 0.0% | 2.3% | 2.3% | 0.0% | 0.0% | 5.9% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 5 | C5      | 34.3% | 12.2% | 2.7% | 10.4% | 34.3% | 8.2% | 3.8% | 2.2% | 10.4% | 6.0% | 4.9% | 2.7% | 2.7% | 1.2% | 0.0% | 0.5% | 0.5% | 0.5% | 0.0% | 0.0% |
| 3 | C6      | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 7 | C7      | 0.0% | 9.3% | 9.3% | 0.0% | 1.9% | 3.7% | 14.8% | 1.9% | 0.0% | 29.6% | 0.0% | 1.9% | 1.9% | 0.0% | 5.6% | 20.4% | 0.0% | 0.0% | 0.0% |
| 3 | C8      | 49.3% | 1.4% | 0.0% | 39.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 5.8% | 0.0% | 0.0% | 0.0% | 4.3% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 5 | C9      | 23.3% | 18.2% | 1.7% | 17.4% | 0.8% | 2.5% | 4.1% | 0.8% | 7.4% | 6.6% | 9.3% | 2.5% | 3.5% | 0.9% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 1.7% |
| 7 | C10     | 9.1% | 9.1% | 59.1% | 4.5% | 0.0% | 4.5% | 0.0% | 0.0% | 0.0% | 0.0% | 4.5% | 0.0% | 9.1% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 3 | C11     | 0.0% | 5.7% | 3.4% | 0.0% | 32.2% | 21.8% | 1.1% | 4.6% | 11.5% | 10.3% | 1.1% | 1.1% | 1.1% | 0.0% | 1.1% | 2.3% | 1.1% | 0.0% | 0.0% |

When \( K = 5 \): C1 (car service): most POI subtypes with the top-level type “Car Maintenance Station” and “Car Service” are found in this cluster; C2 (business and finance): this cluster mainly covers POI top-level types such as “Financial Insurance Service”; C3 (leisure and entertainment): this cluster contains various entertainment places made up of “Sports and Leisure Service” and “Famous Tourist Sites”; C4 (commerce and shopping): commercial type labels such as “Shopping Service” and “Catering Service” can be found in this cluster; C5 (residential community): the distribution of clusters in this region is relatively uniform and the main types are POI types that are closely related to people’s lives such as “Life Service”, “Shopping Service”, and “Healthcare Service”.

When \( K = 7 \): C1 (car service); C2 (business and finance); C3 (leisure and entertainment) C4 (commerce and shopping); C5 (residential community); C6 (transportation service): this cluster mainly contains “Transportation Facility” and some communal facilities that originally belong to C1 and C3; and C7 (science and education culture): some types that originally belong to C5 are separated out, such as “Government Agency” and “Science and Education Service”.

It was found that most of the similar bottom-level types were clustered in the same cluster, so each cluster in the clustering results showed distinct functionality and features, indicating that the POI vectors can effectively show the association between POI types. As we achieved the distributed representation of POIs by capturing their spatial distribution characteristics and the clustering results fully revealed that there was a significant type correlation between the POI vectors of similar spatial distributions, it was reasonable to utilize these POI vectors to explore the spatial and type relevance. As a result, each POI vector does not only represent the type semantics in the form of a point with a type label, but also describes its surrounding environmental characteristics. Therefore, the POI vectors can be used to construct the ROI vectors displaying its regional characteristics and measure the correlation between query conditions and ROI vectors to implement a top-K ROI query.

5.2. ROI Keyword Query Research

The experiment in Section 5.1 shows that the trained POI vectors can reflect the type association and environmental semantics. Next, we evaluated the effectiveness of our method based on the POI vectors for the ROI keyword query on a real dataset in this subsection.

5.2.1. Settings

Dataset

Research region: We selected the main urban area inside the Fifth Ring Road of Beijing as our research region (116.1500°E–116.5969°E, 39.7500°N–40.0563°N in the geographic coordinate system). It was converted into an area with length \( l = 38 \) km and width \( w = 34 \) km after the projection transform. A total of 236,168 POIs with type labels were included in the area, where each of type label was assigned
by its bottom-level type to be used to match the ROI keyword query. Figure 7 shows the distribution of the POIs in our research region.

**Figure 7.** Research region. Yellow lines in the figure denote the main road data of Beijing and the small black dots indicate the POIs of Beijing.

ROI validation set: We obtained vector data of the land use regions and certain special building areas from OpenStreetMap (OSM) [44] and considered them as the verification ROI set to verify the effectiveness of the ROI query. For example, for the original bottom-level type label “university”, there were some corresponding ROIs labeled “university” in our ROI verification set shown in Figure 8.

**Figure 8.** ROI validation set. The figure shows the distribution of the ROIs labeled “university”, represented by the blue regions in the research region. These were utilized to verify the effectiveness of the keyword query for the ROI of the corresponding label.
Query Examples

In order to fully verify the effectiveness of our proposed method, we designed four representative single-keyword query examples for the ROI top-K query and evaluated the query results based on the ROI verification set:

- Q₁ ("industrial park"): The ROIs, whose distribution is concentrated, are mainly located in the suburbs. The area of each single ROI is usually large;
- Q₂ ("university"): The ROIs, whose distribution is concentrated, are mainly located near the city center. The area of each single ROI is usually large;
- Q₃ ("residence community"): The ROIs, whose distribution is dispersed, are located evenly in our research region. Each ROI has a small size of area in general;
- Q₄ ("park"): The ROIs, whose distribution is dispersed, are located evenly in our research region. The area of each single ROI is relatively small.

By testing our methods on ROIs of different scales, the scalability of our approach was verified in a real dataset.

Compared Methods

To demonstrate the effectiveness of our method, two baselines were implemented:

1. Simple Count Query (SCQ): This method counts the number of each POI with bottom-level types in each ROI after constructing the grid division. For the top-K search of the keyword $q$, it returns the top-K ROIs according to the count ranking of the POIs with the corresponding label $t_i$.
2. Dense Query (DQ): An ROI query method based on POI density was proposed in [11]. Considering the effect to the density from adjacent grids, it returned the top-K ROI where the POIs with the corresponding label $t_i$ have a high density for keyword query $q$.

Our method and its variants were considered as follows:

1. ROI2VEC: The ROI vector, which is the mean of the POI vectors in the candidate ROI, is calculated to measure the similarity score with the query vector corresponding to the query keyword $q$ by Equation (10). It returns the top-K ROIs with the highest similarity scores.
2. ROI2VEC + TF-IDF (ROITFIDF): Based on ROI2VEC, the TF-IDF method is used to perform weighted averaging on the POI vectors (Section 4.3.2. TF-IDF Method)
3. ROI2VEC + Gaussian kernel (RGK): Based on ROI2VEC, the Gaussian kernel is used to perform weighted averaging on the candidate ROI vector. (Section 4.3.3. Gaussian kernel)
4. ROI2VEC + TF-IDF + Gaussian kernel (RALL): Add the TF-IDF method and the Gaussian kernel into ROI2VEC.

Parameter Settings

All of the above methods need to set the grid division $a \times b$. In this paper, we used 38 × 34 grids for all comparison experiments. According to the set values, the area of each grid was just about 1 km², which can be accepted to explore the ROI by the user. Parameter analysis will be discussed later. Regarding our method and its variants, to avoid complex computation, only the influence of the neighboring ROIs was taken into consideration. Specifically, the size of the Gaussian kernel was set to $3 \times 3$.

Evaluation Metric

We selected the ROIs corresponding to query $q$ from the ROI validation set and rasterized them on the 38 × 34 grids. Precision, Recall, and F-value, three metrics frequently used in the information retrieval field, were adopted to evaluate the performance of query results. These were defined as
were viewed as the hits, i.e., the correct ROI query results. With the number of top-K query results of the regional characteristics of the ROI. Meanwhile, ROITFIDF, RGK, and RALL improved the query semantic information and environmental information of all POI vectors in the region can reflect more the POIs with keywords, the methods based on ROI2VEC representation that are able to capture the baselines in these query tasks. Compared with the baselines based on the number or the density of regions, which make the query result closer to the actual distribution of the ROIs.

The overlap regions between the top-K results \( R_{\text{top-k}} \) of the query and the rasterized region \( R_v \) were viewed as the hits, i.e., the correct ROI query results. With the number of top-K query results taken as the denominator, \( \text{Precision} \) reflects the proportion of the hits in the top-K query results; with the number of ROIs in the validation set \( R_v \) taken as the denominator, \( \text{Recall} \) reflects the proportion of the hits in all relevant ROI query results. The \( F \)-value is the harmonic average of them. As the \( F \)-value can reflect the overall performance of the query, the \( F \)-value corresponding to the query results was considered as the final evaluation standard in our experiment.

5.2.2. Performance Comparison

We tested each query example by setting the \( K \) value from 10 to 50, with 10 as the step interval and show the experimental results in Figure 9.

\[
\text{Precision} = \frac{\text{count}(R_{\text{top-k}} \cap R_v)}{\text{count}(R_{\text{top-k}})}
\]
\[
\text{Recall} = \frac{\text{count}(R_{\text{top-k}} \cap R_v)}{\text{count}(R_v)}
\]
\[
F = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
\]

The experimental results revealed that our method achieved a better performance than the baselines in these query tasks. Compared with the baselines based on the number or the density of the POIs with keywords, the methods based on ROI2VEC representation that are able to capture the semantic information and environmental information of all POI vectors in the region can reflect more of the regional characteristics of the ROI. Meanwhile, ROITFIDF, RGK, and RALL improved the query

\[\text{Figure 9.} \] The performance comparison of the methods in Q1~Q4, which shows the change of the F-values (y-axis) of the query results with increases in the K value of the top-K query (x-axis).
results of the original ROI2VEC because the TF-IDF method further considers the distribution trait of the number of POIs, and Gaussian kernel takes into account the influence of the surrounding regions, which make the query result closer to the actual distribution of the ROIs.

Faced with the query of an ROI with a concentrated distribution (tasks in large scale, Q1 and Q2), our methods performed better than the baselines, especially the RGK and RALL, since the Gaussian kernel tends to make the results form a connected region, which is effective in the larger ROI exploration. Regarding the query of the small area of ROI with dispersive distribution (tasks in small scale, Q3 and Q4), our methods showed an obvious performance improvement, because the POIs with the type of keywords were evenly scattered in our research regions, so it was difficult for these query tasks to obtain the true characteristics of the ROI only by the count or density of the POIs via keywords. In contrast, though the POIs consistent with the query keyword were evenly distributed, and it was difficult to distinguish the environment of candidate ROIs, our method considered the influence of all POIs on the regional characteristics and environment, which led to better performance. It was noted that the ranges of the F-value among the query tasks were quite different. The F-value not only reflects the Precision of the top-K query, but also reveals the Recall of it. There was a large difference among the areas of the ROIs after the rasterization, which resulted in the difference in Recall and influences the range of the F-value. Similarly, with the increase in the number K of the top-K query, the query results were closer to the original ROI and the Recall rose, which caused an increase in the F-value.

5.2.3. Case Study

Taking Q1 as an example, we specifically analyzed the results of the query task. The rasterization of the original ROI labeled “industrial park” is shown in Figure 10:

![Figure 10. Original ROI rasterization. The labeled ROI occupied 106 grids in total in the research region (38 x 34 grids). The purple grids indicate the labelled regions.](image)

The result of the top-50 query from test methods is shown in Figure 11:

![Figure 11. (a) Simple count query (SCQ) result of the top-50 ROI queries. (b) ROI vector (ROI2VEC) result of the top-50 ROI queries. (c) ROI2VEC + TF-IDF + Gaussian kernel (RALL) result of the top-50 ROI queries.](image)
It was found that the query results of our method were more consistent with the distribution of the original ROI after the visualization, which produced fewer false positives than SCQ. In addition, RALL considers the influence of the surrounding grids to explore the characteristics of connectivity regarding the ROI.

5.2.4. Tuning the Size of the Grids

In the previous experiment, we adopted a grid setting of $38 \times 34$ to make the size of each grid approach $1 \text{ km}^2$. As the grid setting is an important parameter of the proposed methods, we adjusted the size of the grid to test the robustness of our method RALL. Therefore, the different sizes of the grid were set for Q1 as $4 \text{ km}^2$, $1 \text{ km}^2$, $0.25 \text{ km}^2$, and $0.0625 \text{ km}^2$, which corresponded to the division of grids of $19 \times 17$, $38 \times 34$, $76 \times 68$, and $152 \times 136$, respectively. Considering the actual area ratio of the selected ROI in the research region, we set the value of the $K$ in top-$K$ as the 5% of the total number of grids. The results are shown in Figure 12:

![Figure 12. Q1 query results by RALL for the different sizes of the grids. (a) The 4 km$^2$ size of the grid. (b) The 1 km$^2$ size of the grid. (c) The 0.25 km$^2$ size of the grid. (d) The 0.0625 km$^2$ size of the grid. With the same query area ratio set for different tasks, the corresponding F-values were: (a) 0.141, (b) 0.339, (c) 0.297, and (d) 0.215.](image)

The comparison results show that in the settings where the size of the grid was $1 \text{ km}^2$ and $0.25 \text{ km}^2$, there was a higher F-value, which indicates that our method at these scales could better reflect the original ROI. It was found that the results for a larger size setting tended to produce errors and oversize grids could not provide some valuable information and guidance to the users. On the other hand, the grids were so small that the method is similar to the detection of the relevant points and loses the ability to explore the surrounding environment, which results in more errors. In a word, the size of the grid should be based on the user’s knowledge of the ROI.
5.2.5. Time and Space Consumption

It should be noted that most of the time consumption in our method comes from the training process of the POI vectors and the construction of the ROI vector, which can be performed in an off-line manner in advance. In the search step, with the ROI vectors and \( K \) value of top-K given, our method and its variants have the same time complexity \( O\left(\frac{(2n-K+1)K}{2}\right) \) as the simple count query and \( n \) is the number of the grids, which is much larger than \( K \) and directly influences the time complexity. Therefore, the time complexity of our methods was approximately equal to \( O(n) \). While the count method needs to maintain an array whose size is the number (521 in this paper) of the bottom-level types for each grid, our method keeps an array, whose size is the dimension \( D \) (120 in this paper) for each one in memory.

5.2.6. Multi-Keyword Query

Regarding the multi-keyword query, a query group \( Q = \{“Starbucks”, “cinema”\} \) was given to demonstrate the query results by our method. The intention of the query was to explore the related regions to both of the keywords. The original distribution of the POIs is shown in Figure 13.

![Figure 13](image-url)

**Figure 13.** In our research region, the blue POIs represent Starbucks while the yellow POIs represent the cinema. This figure shows their spatial distribution characteristics.

According to their original distribution, a heat map considering the correlation between them is shown in Figure 14a. As an example, the top-50 query results in size 0.25 km\(^2\) were returned by the RALL method, which is shown in Figure 14b.

Compared with Figure 14a, this method was found to be successful in exploring the related ROIs in the map and returning the top-K relevant results based on a correlation meeting the user’s query in Figure 14b. It is worth noting that because the vector of each candidate ROI was prefabricated, the multi-keyword query only adjusted the query vector according to the query keyword group, so that the time complexity of the search step was the same as the single-keyword query, i.e., \( O(n) \).
Finally, we analyzed 6. Conclusions

and preferences, with the understanding of regional environmental semantics. and similar ROI recommendations by considering the user's personal information, historical visits, improve the performance of ROI exploration. Another direction worth exploring is to make interesting human activities, such as check-in data related to LBSs and mobile phone location data, to further

environmental characteristics and semantic information of the POIs, which means that the applicable granularity of ROI, which a

acquire the similarity score with the vectorized keyword query to implement the ROI top-K search. In this paper, we proposed a novel ROI exploration method, with a distributed representation of the POI, that considered the environmental information inside the region by learning its internal POI embedding vectors and calculating the corresponding candidate ROI vectors, which were utilized to acquire the similarity score with the vectorized keyword query to implement the ROI top-K search. First, we improved the construction of the POI corpus and proposed a more reasonable POI embedding method. As a result, the validity of the acquired POI vector was verified by the established evaluation metric after discussing the relationship between the quality of them and the parameter selection. Next, compared with the baselines on a real large-scale dataset, the experimental results showed that our method achieved a significant improvement in the performance of ROI exploration, reflecting the precious value of environmental semantics for spatial region exploration tasks. Finally, we analyzed the time and space consumption of the proposed method and achieved an expansion of multi-keyword ROI queries.

Two limitations of our method need to be clarified: (1) The size of the grid determines the query granularity of ROI, which affected the performance of our proposal. Unfortunately, we were not able to automatically learn this value based on the target of the query, which means that users need to set it up based on experience; and (2) The essence of the distributed representation of the POI is to learn the environmental characteristics and semantic information of the POIs, which means that the applicable objects of our method will depend on the cities’ schemas that constitute the POI corpus. An intuitive example from POI vectors learned from Beijing might be efficient to build a spatial keyword query of ROI in Shanghai but might lead to bad performance in rural towns.

In the future, we will attempt to integrate more novel mobility data sources closely related to human activities, such as check-in data related to LBSs and mobile phone location data, to further improve the performance of ROI exploration. Another direction worth exploring is to make interesting and similar ROI recommendations by considering the user’s personal information, historical visits, and preferences, with the understanding of regional environmental semantics.

Figure 14. (a) The heat map of the POIs. It intends to reflect a combined relevance of the POIs of the type of Starbucks and cinema. Brighter grids denote a higher value of their combined relevance, i.e. both are densely distributed in this ROI, while the dark ones are the opposite. It is worth noting that the grids populated by only one type of POI do not show a very high correlation. (b) The top-50 query results by RALL. The top-50 query results are basically consistent with the brighter grids in (a), reflecting that our method can achieve good performance in the task of multi-keyword queries.
Author Contributions: Xiangdian Zhu and Ye Wu designed and implemented the algorithm; Xiangdian Zhu and Ye Wu performed the experiments and analyzed the data; Luo Chen and Ning Jing contributed to the construction of experimental environment; Xiangdian Zhu wrote the paper; and Ye Wu and Luo Chen helped to improve the language expression.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 41871284 and the National Natural Science Foundation of Hunan Province, grant number 2019JJ50718.

Conflicts of Interest: The authors declare no conflict of interest.

References


19. Cary, A.; Wolfson, O.; Rishe, N. Efficient and scalable method for processing top-K spatial boolean queries. SSDM 2010, 6187, 87. [CrossRef]


© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).