Clue-based Spatio-textual Query

Junling Liu
Northeastern University, China
Shenyang Jianzhu University, China
liujl@sjzu.edu.cn

Ke Deng RMIT University, Australia
ke.deng@rmit.edu.au

Huanliang Sun
Shenyang Jianzhu University, China
sunhl@sjzu.edu.cn

Yu Ge
Northeastern University, China
yuge@mail.neu.edu.cn

Xiaofang Zhou
The University of Queensland, Australia
Soochow University, China
zxf@itee.uq.edu.au

Christian S. Jensen
Aalborg University, Denmark
csj@cs.aau.dk

ABSTRACT

Along with the proliferation of online digital map and location-based service, very large POI (point of interest) databases have been constructed where a record corresponds to a POI with information including name, category, address, geographical location and other features. A basic spatial query in POI database is POI retrieval. In many scenarios, a user cannot provide enough information to pinpoint the POI except some *clue*. For example, a user wants to identify a café in a city visited many years ago. SHe cannot remember the name and address but she still recalls that "the café is about 200 meters away from a restaurant; and turning left at the restaurant there is a bakery 500 meters away, etc.". Intuitively, the clue, even partial and approximate, describes the spatio-textual context around the targeted POI. Motivated by this observation, this work investigates *clue-based spatio-textual query* which allows user providing clue, i.e., some nearby POIs and the spatial relationships between them, in POI retrieval. The objective is to retrieve k POIs from a POI database with the highest spatio-textual context similarities against the clue. This work has deliberately designed data-quality-tolerant spatio-textual context similarity metric to cope with various data quality problems in both the clue and the POI database. Through crossing valuation, the query accuracy is further enhanced by *ensemble method*. Also, this work has developed an index called *roll-out-star R-tree* (RSR-tree) to dramatically improve the query processing efficiency. The extensive tests on data sets from the real world have verified the superiority of our methods in all aspects.

1. INTRODUCTION

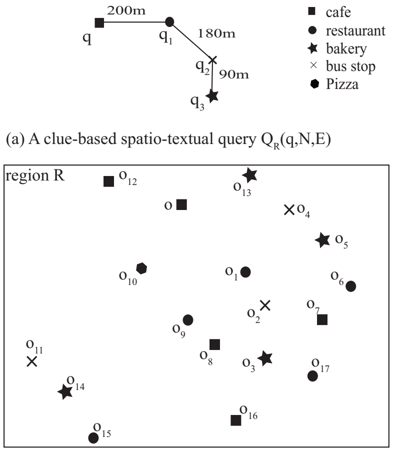
Along with the proliferation of online digital map and location-based service, very large POI (point of interest) databases have been constructed, for example, World POI Database (mcrenox.com.ar/worldpoidb) contains about 5 million POIs where each record of POI consists of name, category, address, geographical location and other features. POI search is one of the basic queries in POI databases. It is commonly run by providing exact address which can uniquely pinpoint the location, e.g., "124 La Trobe St, Melbourne"; or specifying distinguishable category (i.e., only few POIs with this category) in a region, e.g., "post office, Melbourne". However, it is not uncommon that a user queries for a POI in a city, whose category is less distinguishable such as café, but she can provide more or less spatio-textual context information around the targeted POI. Consider a scenario in our daily life: a user wants to identify a café in a city visited many years ago. She cannot remember the shop name and address, but she still recalls that "the café is about 200 meters away from a restaurant; and turning left at the restaurant there is a bakery 500 meters away, etc.". The information usually cannot exactly locate a POI but intuitively it provides clue to help identify the most promising POIs.

Motivated by this observation, this work studies a novel query type named clue-based spatio-textual query which allows user providing clue (or called spatio-textual clue) in POI retrieval. Clue is the user-provided information which specifies the spatio-textual context of the querying POI including the nearby POIs and the spatial relationships between them. The clue-based spatio-textual query is denoted as $Q_R(q, N, E)$ where $q \in N$ is the querying POI, $N \setminus \{q\}$ is the set of clue POIs, E is the set of edges between POIs in N, and R is the region where the POIs in N are located. Given a POI database, the query objective is to identify k POIs which have the same category as q and have the highest spatio-textual context similarities against the clue. **Fig.** 1 (a) and (b) illustrate a clue-based spatio-textual query and the POIs in region R in a POI database.

The clue-based spatio-textual query is challenged by the data quality problems in both the clue and the POI databases.

- *POI database update/incompleteness*: While more and more businesses/venues have records in POI databases today, it is unsurprising that many businesses/venues have not been recorded. On the other hand, a POI database evolves over time as the response to the change of the real world. It causes the following problems: (**i.**) the category of a POI may have been changed to another category; (**ii.**) a POI may have been deleted from the databases.
- *Partial/approximate information of clue*: The clue is typically based on personal observations such that: (**i.**) the clue POIs are a random subset of objects in the proximity around the targeted POI; (**ii.**) the category of POI in the clue may be incorrect (e.g., a user may think of a POI *post office* being

Proceedings of the VLDB Endowment, Vol. 10, No. 5
Copyright 2017 VLDB Endowment 2150-8097/17/01.



(a) A clue-based spatio-textual query $Q_R(q,N,E)$

(b) POIs in D in the region R

Figure 1: An example of clue-based spatio-textual query.

news-agency) or inexact (e.g., a user may think of a POI being *diner* whereas in the POI database it may be referred to as a *restaurant*); (**iii.**) the spatial relationships between POIs are approximate, this is a natural phenomenon for human to estimate distance/direction, for example, if the distance between two POIs in the clue is about 100 meters, the actual distance may be noticeably greater than or less than 100 meters, so are the directions.

Given a clue-based spatio-textual query $Q_R(q, N, E)$, suppose o in the database is the truly matching POI of the querying POI q. Due to the data quality problems (i.e., POI category incorrect in the clue, POI missing or category update in the databases), some clue POI $q_i \in N \setminus \{q\}$ may not have truly matching POI in the databases; on the other hand, even though q_i does have the truly matching POI, say o_i, in the database, o_i may be away from the location specified by the clue. To attack such problems, this work has developed data-quality-tolerant spatio-textual context similarity metric. The scheme is featured by partial matching where the well matched POIs, maybe a small fraction of all POIs in the clue, will lead to high similarity and the poorly matched POIs are strategically considered because of data quality problems and thus ignored largely.

An even more serious issue caused by the data quality problems is that the truly matching POI o of the querying POI q does not exist in the databases. The challenge is that this situation is unknown in advance. So, we further investigate an *ensemble method*. In this method, a number of new queries based on the original query $Q_R(q, N, E)$ are generated. Specifically, for each clue POI $q_i \in N \setminus \{q\}$, we generate a new query $Q_R(q_i, N, E)$ where q_i is treated as the querying POI, the original query POI q is treated as a clue POI. As a result, even though the original querying POI does not have truly matching POI in the databases, we still have chance to retrieve the truly matching POIs of the clue POIs. Since the spatial relationships between POIs have been specified by the clue, the location of the truly matching POI of q can then be inferred. More importantly, given a clue-based spatio-textual query, the ensemble method tends to provide more reliable solution because the results of generated queries are cross-validated and the ones agreed by the most are returned.

Since the category of a POI in the query maybe be inexact, we extend the category with thesaurus (thesaurus.com). Specifically, the category of each POI in the clue is extended to include the thesaurus which are interchangeable terms of the original category. For example, the interchangeable terms of *restaurant* including *bar*, *cafeteria*, *coffee shop*, *diner*, and *dining room*.

Even though data quality problems are inevitable, the clue-based spatio-textual query assumes that the information in the query is largely correct. In addition, we assume that the POI databases have recorded numerous businesses/venues and the update of POI database is infrequent. The assumption is reasonable for a real world application. Nowadays, the large POI databases have been established and continuously grow. Meanwhile, the POI database updates infrequently since it corresponds to the real-world objects which change over time but slowly.

The contributions of this work are threefold:

1. The clue-based spatio-textual query enables user to explore spatio-textual context information in POI retrieval. Even though the spatio-textual context information is proverbial in practice, to the best of our knowledge, it has not been effectively utilized previously.
2. The data-quality-tolerant spatio-textual context similarity metric has been proposed to identify the truly matching POI even though various data quality problems present in both the clue and the POI databases. Moreover, the ensemble method has been studied to provide more robust solution.
3. Powerful query processing methods have been investigated. In particular, a novel index called *roll-out-star R-tree* (RSR-tree) has been proposed to lift the query processing efficiency.

More interestingly, the clue-based spatio-textual query can be found in a wider spectrum of applications where the spatio-textual context similarity is concerned. Here are two examples. First, the landscape design around a building includes the positions of small ponds, trees, car parks, sport grounds, etc.; given a design, it is critical to retrieve the previous designs with similar layout for the purpose such as design verification or intelligent property protection. Second, sport such as American Football, Football/Soccer, Basketball, Hockey, and Rugby is full of attack, defence and tactic. The positions of different players of both sides around the player organizing the attack or defence decide the tactic to be applied. It is important to learn from history by retrieving the matches with the similar position layout for tactic analysis [1].

The rest of this work is organized as followings. First, Section 2 reviews the related work and the problem is defined in Section 3. Then, a data-quality-tolerant spatio-textual context similarity metric is introduced in Section 4. After that, Section 5 presents the query processing algorithm, and Section 6 proposes an index called roll-out-star R-tree to lift query processing efficiency. The ensemble method is investigated in Section 7 and the experimental results are reported in Section 8. Finally, this work is concluded in Section 9.

2. RELATED WORK

Spatio-textual Query: The spatio-textual similarity join has been investigated [1, 2]. A spatio-textual object o consists of location $o.loc$ and textual information $o.text$. $o.text$ is a set of words (or tokens) each of which could carry a weight modelling the relevance to o. For every pair of spatio-textual objects o_i and o_j, the spatial distance $dist(o_i, o_j)$ is measured using the Euclidean distance and the textual similarity $sim_t(o_i, o_j)$ is measured using Jaccard coefficient (i.e., the intersection of the words associated with o_i and o_j divided by their union). Given a spatial distance threshold ϵ and a textual similarity threshold θ, the aim is to retrieve all pairs (o_i, o_j)

[1] outsideoftheboot.com

	Signature
basketball	010001001
cooking	000101001
hunting	100101000
Employee	110101001

(a)

	Signature
basketball	010001001
cooking	000101001
fishing	011001000
Query	011101001

(b)

Figure 2: Signature.

such that $dist(o_i, o_j) \le \epsilon$ and $sim_t(o_i, o_j) \ge \theta$. In [3], Hu et al. investigate a variant of the spatio-textual similarity join where the spatial distance and textual similarity are aggregated to an overall spatio-textual similarity. Given two sets of spatio-textual objects S and R, the query aims to identify k pairs from $S \times R$ with the highest overall spatio-textual similarities.

Given a query q that specifies a set of words and spatial information, the problem to search for the objects with high spatio-textual similarity with respect to q has been studied by [4, 5, 6, 7]. In [6], the query consists of a specified region $q.R$ and a set of words $q.T$. Given a set of objects $O = \{o_1, \cdots, o_n\}$, each $o_i \in O$ is defined as $(o_i.R, o_i.T)$ where $o_i.R$ is the region of o_i and $o_i.T$ is the set of words associated with o_i. Given q and o_i, the spatial similarity is measured by the overlap of their regions and the textual similarity is measured by the overlap of their words. The query returns a subset of objects $A \subseteq O$ such that, for each $o \in A$, the spatial similarity with respect to q is greater than a threshold τ_R and the textual similarity with respect to q is greater than a threshold τ_T. In [4, 5, 7], the variants of the problem have been investigated with different spatio-textual similarity metrics. In addition, the clustering techniques based on spatio-textual relevance have been investigated [8].

Our clue-based spatio-textual query differs from the existing spatio-textual queries. We concern the spatio-textual context similarity based on distance and relative directions between POIs with same textual information (i.e., category). In particular, the relative direction is irrelevant to any existing spatio-textual query while it plays the key role in the clue-based spatio-textual query. Simply extending the techniques developed for existing spatio-textual queries does not solve our problem.

Signature Index: Signatures are hash coded binary words of fixed length derived from objects stored in the database (see [9] for a complete survey). They present abstractions of objects. Signatures can provide a filter for testing attribute inclusion for objects, because if the subset condition does not hold for the signature, then it does not hold for the object neither. For example, consider the query "find all employees who like to spend their free time with *cooking*, *fishing*, *basketball*". Suppose that an employee's hobbies are "*basketball*, *cooking*, *hunting*". In **Fig.** 2 (a)(b), the signatures of hobbies and their superimposition are illustrated. As shown, since the query signature is not a subset of the employee signature, the specific employee cannot be part of the answer to the query. In [7], it proposes IR^2-tree which is a combination of an R-Tree and signature files. In particular, each node of an IR^2-tree contains both spatial and textual information; the former in the form of a minimum bounding region and the latter in the form of a signature. A leaf node has entries of the form $(ObjPtr, A, S)$ where $ObjPtr$ is the pointer referring to a spatial object, say o, A is the location of o and S is the signature of textual information associated with o. A non-leaf node has entries of the form $(NodePtr, A, S)$ where $NodePtr$ is the pointer referring to a child node, A is the minimum bounding region of all objects in the child node, and S is the superimposition of all signatures of all objects in the child node.

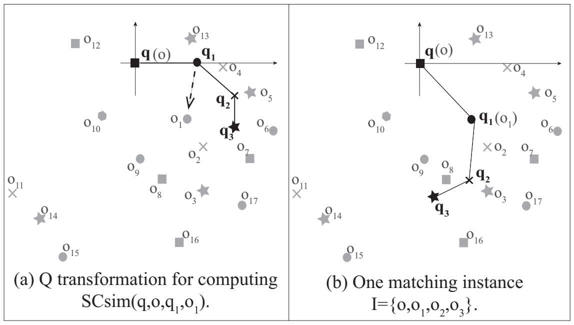
In this work, the spatio-textual context of an object o is represented by a roll-out-star which can be viewed as the signature of o. While the existing signatures represent textual information only, the roll-out-star concurrently represents the textual information as well as the distance and relative directions of the textual informations in space. The proposed roll-out-star R-tree (RSR-tree) has the similar structure as the IR^2-tree. However, the node evaluation in RSR-tree involves complex geometric operation while in IR^2-tree involves bitwise operations only.

Spatial Scene Query: A spatial scene comprises a collection of spatial objects and their particular arrangement. For example, X is an *academic building* and Y, Z are *parking places*; X meets Y, Y meets Z and X disjoints Z. The central question of spatial-scene query is how to associate the objects and relations of one scene in a spatial scene query with the corresponding objects and relations of another scene, typically a sub-scene of a map in databases [10, 11]. For a query graph G with node set $(v_1, \cdots, v_n)$ and a database graph H with node set $(u_1, \cdots, u_m)$, spatial scene query execution takes two stages. The first stage creates an association graph A where each node corresponds to each compatible pair of nodes between G and H and an edge is inserted between nodes of A if their corresponding nodes in H have the same relationship as that in G. In the second stage, the maximum-maximal cliques are identified as the complete solution (i.e., all query objects and relations have a counterpart in the databases), and the maximal but not maximum cliques are identified as incomplete solution (i.e., a subset of all query objects and relations have a counterpart in the databases). The relationships between objects concerned in spatial scene query are topological relationships including *disjoint*, *meet*, *equal*, *overlap*, *inside*, *contains*, *covers*, *coveredBy* and the relationships derived from them.

In the clue-based spatio-textual query, all objects are POIs which are disjoint to each other by the definition of spatial scene query in [10, 11]. Moreover, the distances and relative directions between POIs concerned by the spatio-textual context similarity cannot be evaluated by the methods developed by [10, 11].

Spatial/distance Join: The simplest form of spatial join is defined on two sets of objects, and retrieves a subset of the Cartesian product of these two sets, filtered by the spatial predicate, which is a spatial relationship between the objects in the result [12]. A typical example is "*find all cities that are crossed by a river*". The distance join is the spatial join by extending predicate *overlap* to predicate *within* which can answer the query like "*find all cities within 20 miles of forests*" [13]. In [13], Hjaltason et al. have developed algorithms to compute distance joins efficiently by traversing two indexes, each for one set of objects. In [14], the *multiway spatial joins* have been investigated, which is an extended version of spatial join involving more than two spatial object sets, for example "*find all cities within 20 miles of forests and crossed by a river*". Multiway joins output objects, each from one data set, satisfying the predicates which represent the pairwise relationship of different sets. In the spatial/distance joins, the spatial relationship between objects are exact without considering data quality problems. So, they are incapable for processing clue-based spatio-textual query even though the clue-based spatio-textual query aims to search for a set of objects with particular pairwise spatial relationships as well.

Uncertain/fuzzy Spatial Query: In [15], Cheng et al. have proposed the probabilistic nearest-neighbour query (PNNQ) in uncertain environments, which finds a set of data objects that have non-zero probability to be the nearest neighbour of the query point. In [16], Tao et al. have developed U-tree to index multi-dimensional uncertain data with arbitrary probability density functions (PDF). In [17], Zheng et al. have indicated that a fuzzy object



(a) Q transformation for computing SCsim(q,o,q_1,o_1).

(b) One matching instance I={o,o_1,o_2,o_3}.

Figure 3: Q transformation.

is often represented by a PDF, and the accumulation of probabilities within a PDF region is equal to one. The uncertain data model disobeys this assumption. An index method suitable for fuzzy data nearest neighbour query has been studied in [17]. In clue-based spatio-textual query, the clue and the POI databases have multiple data quality problems but they are irrelevant to uncertainty/fuzzy data. The reason is that the data in the clue and in the POI databases are not with any probability to indicate how likely they are true. So, the techniques specialized for processing uncertain/fuzzy data are irrelevant to the clue-based spatio-textual query.

3. PROBLEM DEFINITION

The POI database is denoted as D. Each POI $o \in D$ is represented as $(o.id, o.loc, o.cid)$ where $o.id$ is the identity of o, $o.loc$ refers to the location of o, and $o.cid$ is the category identity to indicate o is, for example, a *café* or a *restaurant*.

Definition 1 (Clue). *When querying a POI in a POI database, clue is the user-provided information which specifies the spatio-textual context of the querying POI. It includes the categories of nearby POIs around the querying POI, called clue POIs, and the spatial relationships (i.e., distances and relative directions) between them (including the querying POI and clue POIs).*

A clue-based spatio-textual query is denoted as $Q_R(q, N, E)$ where q is the querying POI, $N \setminus \{q\}$ is the set of clue POIs, E is the set of edges which represent the relative spatial relationships between the POIs in N, and R is the region where the POIs in N are located. Each POI $q_i \in N$ is with a category identity, denoted as $q_i.cid$. Each edge $e \in E$ is represented as $(e.id, e.q_i, e.q_j, e.dist, e.dir)$ where $e.id$ is the edge identity, $e.q_i$ and $e.q_j$ indicate the two end POIs of e, $e.dist$ is the distance between $e.q_i$ and $e.q_j$, and $e.dir$ is the direction of e. In particular, the direction of one edge is defined as 0^o and the direction of any other edge is relative to it. In a clue-based spatio-textual query, one does not assume absolute direction (such as East) [2]. The clue-based spatio-textual query can be extracted from a sketch of a graph as depicted in [11].

Let $D_R(q.cid)$ be the POIs in D with the same category as q in region R. Given a clue-based spatio-textual query $Q_R(q, N, E)$ and any POI $o \in D_R(q.cid)$, the spatio-textual context similarity between q and o is denoted as $SCsim(q, o)$. Specifically, $SCsim(q, o)$ assesses to which extent o has the similar spatio-textual context as q.

Table 1: Summary of Notations

Notation	Explanation
D	A POI database.
D_R	The POIs in D in region R.
$*.cid$	The category of POI $*$.
$D_R(q_i.cid)$	The POIs in D with category $q_i.cid$ in region R.
$Q_R(q, N, E)$	A clue-based spatio-textual query.
γ	The scale factor of Q transformation.
$SCsim$ (q, q^m, o, o^m)	The spatio-textual context similarity between o and q in the case that q, q^m truly match o, o^m respectively.
$SCsim(q, o)$	The spatio-textual context similarity between o and q.
$o.RS$	Roll-out-star of POI $o \in D$.
$gr(q_i)$	The grey region of POI q_i.
$gc(q_i)$	A grid cell in $o.RS$ with category $q_i.cid$.
τ_i	The number of edges linked with POI q_i in the query.
RSR-tree	Roll-out-star R-tree.
$A \setminus B$	The set of elements in A but not in B.

Definition 2 (clue-based spatio-textual query). *Given a POI database D, a clue-based spatio-textual query $Q_R(q, N, E)$ retrieves k POIs, $A \subseteq D_R(q.cid)$, such that*

$$SCsim(q, o_i) > SCsim(q, o_j),\ o_i \in A,\ o_j \in D_R(q.cid) \setminus A. \tag{1}$$

The notations used in this work are summarized in Table 1.

4. SPATIO-TEXTUAL CONTEXT SIMILARITY

The spatio-textual context similarity must be data quality tolerant. Desirably, if a POI $o \in D_R(q.cid)$ is the truly matching POI of the querying POI q, $SCsim(q, o)$ should be significantly high even though data quality problems present. We first introduce a concept called *minimum matching requirement*. If this minimum requirement is unsatisfied, it is too weak to justify the match between q and o.

Definition 3 (Minimum Matching Requirement). *Given a clue-based spatio-textual query $Q_R(q, N, E)$, if a POI $o \in D_R(q.cid)$ truly matches the querying POI q, the minimum matching requirement is that at least one clue POI $q^m \in N \setminus \{q\}$ has truly matching POI o^m with the same category as q^m in $D_R \setminus \{o\}$.*

If q truly matches o and one POI $q^m \in N \setminus \{q\}$ truly matches one POI $o^m \in D_R(q^m.cid) \setminus \{o\}$, for each remaining POI $q_i \in N \setminus \{q, q^m\}$, we find the most promising matching POI $o_i \in D_R(q_i.cid) \setminus \{o, o^m\}$. If o_i and q_i match well, i.e., spatially close, it supports that o, o^m truly match q, q^m respectively. To implement such spatio-textual context similarity metric, we place q at the same location as o, and then transform $Q_R(q, N, E)$ in order to place q^m at the same location as o^m. This operation is called Q *transformation* which rotates and uniformly scales $Q_R(q, N, E)$. In specific, the uniform scaling is a linear transformation that enlarges or shrinks $Q_R(q, N, E)$ by a scale factor γ that is same in all directions transformed. The formulas of Q transformation are presented in Appendix-A [18].

[2] The absolute direction is additional information which can help reduce the search space. This work provides solutions to the general problem setting where only relative spatial relationships between POIs are given in the clue. However, our solution can be straightforwardly extended to process absolute directions if they are available.

As shown in **Fig.** 3 (a), q is placed at the same location as o for computing the spatio-textual context similarity between q and o. After that, q_1 and o_1 are selected as q^m and o^m respectively. The Q transformation is performed as shown in **Fig.** 3 (b) such that q_1 is placed at the same location as o_1. It means that o, o_1 truly match q, q_1 respectively. Then, q_2 may match o_2, o_4, o_{11} which are POIs in $D_R(q_2.cid)$ and q_3 may match o_3, o_5, o_{13}, o_{14} which are POIs in $D_R(q_3.cid)$.

DEFINITION 4. *(Matching Instance) Given $Q_R(q, N, E)$ and $o \in D_R(q.cid)$, one POI in $N \setminus \{q\}$ is selected as q^m and one POI in $D_R(q^m.cid) \setminus \{o\}$ is selected as o^m. After Q transformation, q, q^m are at the same locations as o, o^m respectively. A matching instance of $N = \{q, q^m, q_1, \cdots, q_n\}$ is a subset of POIs in D_R, denoted as $I = \{o, o^m, o_1, \cdots, o_n\}$, where q, q^m match o, o^m respectively, q_i matches o_i for $1 \leq i \leq n$, $q_i.cid = o_i.cid$, and $o_i.id \neq o_j.id$ if $i \neq j$.*

For example in **Fig.** 3 (b), $N = \{q, q_1, q_2, q_3\}$ may have many matching instances including $\{o, o_1, o_2, o_3\}$, $\{o, o_1, o_4, o_{13}\}$, $\{o, o_1, o_2, o_3\}$ and $\{o, o_1, o_4, o_5\}$. The spatio-textual context similarity between q and o, in the situation that q^m truly matches o^m, is defined as follows:

$$SCsim(q, o, q^m, o^m) := \gamma \max_{I \subseteq \Phi} SCsim(N, I). \qquad (2)$$

where Φ is the set of all possible matching instances of N in D_R.

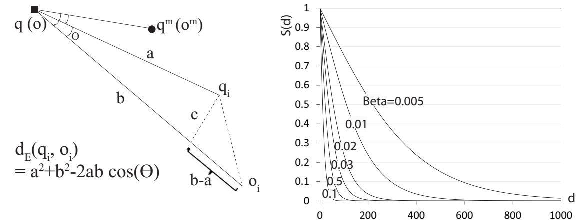
$$SCsim(N, I) := \sum_{q_i \in N, o_i \in I} \tau_i * S(d_E(q_i, o_i)).$$

$$\gamma = S(|d_E(q^m, q) - d_E(o^m, o)|).$$

$$S(d) = 2 - 2 * \frac{1}{1 + e^{-d\beta}}.$$

Where $o_i \in I$ corresponds to $q_i \in N$, $d_E(o_i, q_i)$ is the Euclidean distance between POI o_i and POI q_i, τ_i is the number of edges linked with POI q_i in the clue, $S(d)$ is a sigmoid function to normalize distance d, and γ is the scale factor of Q transformation. In particular, $d_E(q_i, o_i)$ characterises the difference between q_i and o_i in terms of their distances to q as well as their relative directions against edge (q, q^m). Note q, q^m are at the same locations as o, o^m respectively. As shown in **Fig.** 4 (a), θ is the difference between angle $\angle q_i q q^m$ and angle $\angle o_i q q^m$; $d_E(q, o_i) = b$ and $d_E(q, q_i) = a$. If $(b - a)$ is fixed, the greater θ leads to the greater c, and the greater a leads to the greater c. That is, if a clue POI is closer to the querying POI, the directional information is less important in $d_E(q_i, o_i)$; otherwise, the directional information is more important in $d_E(q_i, o_i)$.

As shown in **Fig.** 4 (b), the parameter β in sigmoid function adjusts the slope of the curve. When $d_E(q_i, o_i)$ is smaller, the value of sigmoid function is closer to 1; otherwise, it is closer to 0. Note if γ is very small, it means that it is very unlikely q^m truly matches o^m because $d_E(q^m, q)$ is very different from $d_E(o^m, o)$. For example in **Fig.** 3 (b), we compute the spatio-textual context similarity between $N = \{q, q_1, q_2, q_3\}$ and $I = \{o, o_1, o_2, o_3\}$. Since q and q_1 are at the same locations as o and o_1 respectively, $d_E(q, o) = 0$ and $d_E(q_1, o_1) = 0$. The Euclidean distance between o_2 and q_2 is $d_E(q_2, o_2)$ and the Euclidean distance between o_3 and q_3 is $d_E(q_3, o_3)$. So, $SCsim(Q_R(q, N, C), I)$ is $\gamma(3 * S(0) + 2 * S(d_E(q_2, o_2)) + S(d_E(q_3, o_3)))$ where $\gamma = S(d_E(o, o_1) - d_E(q, q_1))$. A case study shows the effectiveness of spatio-textual context similarity metric in Appendix-B [18].



(a) Role of distance and direction

(b) The sigmoid function

Figure 4: The similarity metric.

DEFINITION 5 (SPATIO-TEXTUAL CONTEXT SIMILARITY). *The spatio-textual context similarity between q and o is defined as*

$$SCsim(q, o) = \max_{q^m \in T1,\ o^m \in T2} SCsim(q, o, q^m, o^m). \qquad (3)$$

where $T1 = N \setminus \{q\}$ and $T2 = D_R(q^m.cid) \setminus \{o\}$.

The category of a POI in the query may be inexact. The effective method to handle this situation is to establish controlled vocabularies, such as taxonomy, thesaurus and ontology, which contain terms with their synonyms and alternative spellings combined to form concepts [19]. A taxonomy is the simplest variant as it contains only terms that are organized into a hierarchical structure. A thesaurus adds non-hierarchical relationships between concepts and other properties to each concept. Ontologies are on the heavy end of the spectrum. Ontologies can express axioms and restrictions. The controlled vocabulary is out of scope of this paper. In this work, without loss of generality, the thesaurus (according to thesaurus.com) is used. Specifically, the category of each POI in the query is extended to include the thesaurus which are interchangeable terms of the original category. For example, the interchangeable terms of *restaurant* including *bar*, *cafeteria*, *coffee shop*, *diner*, and *dining room*.

5. QUERY PROCESSING ALGORITHM

The clue-based spatial query processing is outlined in Algorithm 1. For each POI $o \in D_R(q.cid)$, the spatio-textual context similarity to q, $SCsim(q, o)$, is computed. The $SCsim(q, o)$ computation is based on computing $SCsim(q, o, q^m, o^m)$ for all pairs of $q^m \in N \setminus \{q\}$ and $o^m \in D_R(q^m.cid) \setminus \{o\}$ (line 3-8). The algorithm returns the POI in $D_R(q.cid)$ with the highest spatio-textual context similarity. This algorithm can be directly extended to searching for k POIs in $D_R(q.cid)$ with the highest spatio-textual context similarities.

$SCsim(q, o, q^m, o^m)$ in line 5 is defined in Equation (2). Given q, q^m, o and o^m, there are many matching instances. Since the number of matching instances is potentially large, it is computationally expensive to process them one by one. We propose an efficient algorithm for computing $SCsim(q, o, q^m, o^m)$ which is outlined in Algorithm 2. First, $Q_R(q, N, E)$ has been transformed such that q, q^m are at the same locations as o, o^m respectively. Then, for each POI $q_i \in N \setminus \{q, q^m\}$, the spatially closest POI in $D_R(q_i.cid) \setminus \{o, o^m\}$, denoted as o_{imax}, is identified. For all POIs in N, $\{q, q^m, q_1, \cdots, q_n\}$, we obtain $\Lambda = \{o_{max}, o^m_{max}, o_{1max}, \cdots, o_{nmax}\}$ where q, q^m match o_{max}, o^m_{max} respectively and q_i matches o_{imax} for $1 \leq i \leq n$. If all items in Λ are distinct, Λ is the matching instance leading to $SCsim(q, o, q^m, o^m)$.

Let $N(cid)$ denote the POIs with category cid in N. Suppose $N(cid) \setminus \{q, q^m\}$ contains more than one POIs. For each

Algorithm 1: Clue-based spatio-textual query Processing Algorithm

```
Input: $Q_R(q, N, E)$, $D$.
Output: The POI $o \in D_R(q.cid)$ with the most
        similar spatio-textual context as
        $q$.
for each POI $o \in D_R(q.cid)$ do
    $temp \leftarrow$ 0;
    for each $q^m \in N \setminus \{q\}$ do
        for each $o^m \in D_R(q^m.cid) \setminus \{o\}$ do
            $value \leftarrow SCsim(q, o, q^m, o^m)$;
            if $temp < value$ then
                $temp \leftarrow value$;
                $cur \leftarrow o$;
return $cur$;
```

Algorithm 2: $SCsim(q, o, q^m, o^m)$ Algorithm

```
Input: $Q_R(q, N, E)$, $D$, $o$, $q^m$, $o^m$.
Output: $SCsim(q, o, q^m, o^m)$.
1  $\Lambda \leftarrow \setminus\{o, o^m\}$;
2  for each POI $q_i \in N \setminus \{q, q^m\}$ do
3      $o_{imax} \leftarrow$ the spatially closest POI in
       $D_R(q_i.cid)$;
4      $\Lambda \leftarrow \Lambda \cup o_{imax}$;
5  if POIs in $\Lambda$ are distinct then
6      return $\gamma \sum_{q_i \in N, o_i \in \Lambda} \tau_i * S(d_E(q_i, o_i))$;
7  else
8      for each category cid in $\Lambda$ do
9          $\Lambda(cid) \leftarrow$ POIs in $\Lambda$ with category $cid$;
10         if POIs in $\Lambda(cid)$ are not distinct then
11             $\Lambda \leftarrow \Lambda \setminus \Lambda(cid)$;
12             $\Lambda(cid) \leftarrow AugPath(Q(cid), D_R(cid))$;
13             $\Lambda \leftarrow \Lambda \cup \Lambda(cid)$;
14     return $\gamma \sum_{q_i \in N, o_i \in I} \tau_i * S(q_i, o_i)$;
```

POI in $N(cid) \setminus \{q, q^m\}$, the closest POI in $D_R(cid)$ is maintained in Λ. It is possible that the closest POIs of different POIs in $N(cid) \setminus \{q, q^m\}$ are identical. In this situation, some items in Λ are identical. Clearly, such Λ is not a matching instance of the query because it is not a one-to-one match with N. To address this issue, a further process is needed. For the POIs in $N(cid) \setminus \{q, q^m\}$ and the POIs in $D_R(cid) \setminus \{o, o^m\}$, we apply the Augmenting Path algorithm which is one of the Maximum Bipartite Matching algorithms [20] (line 12 in Algorithm 2). The aim is to maximize the $\sum_{q_i \in T3, o_i \in T4} \tau_i * S(d_E(q_i, o_i))$ where $T3 = N(cid) \setminus \{q, q^m\}$ and $T4 = D_R(cid) \setminus \{o, o^m\}$. The running time of Augmenting Path algorithm is $\mathbf{O}(n_1 n_2)$ where n_1 is the number POIs in $N(cid) \setminus \{q, q^m\}$ and n_2 is the number of POIs in $D_R(cid) \setminus \{o, o^m\}$. Since the number of POIs in the same category in the query is generally very small, the processing cost is trivial.

LEMMA 1.

$$SCsim(q, o, q^m, o^m) := \gamma \sum_{q_i \in N, o_i \in \Lambda} \tau_i * S(d_E(q_i, o_i)). \quad (4)$$

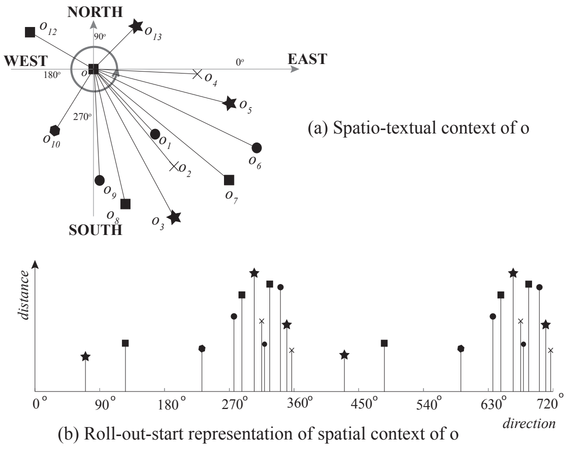
PROOF. In the case that all items in Λ are distinct, the distance between each POI in $q_i \in N$ and the matching POI $o_i \in \Lambda$ is the minimum. So, no other matching instance I may have greater $\sum_{q_i \in N, o_i \in I} \tau_i * S(d_E(q_i, o_i))$ than that of Λ. In the case that all items in Λ are not distinct, $\sum_{q_i \in T3, o_i \in T4} \tau_i * S(d_E(q_i, o_i))$ is maximized for every category in the query. So, no other matching instance I may have greater $\sum_{q_i \in N, o_i \in I} \tau_i * S(d_E(q_i, o_i))$ than that of Λ. The lemma is proved. □

6. SPATIO-TEXTUAL CONTEXT INDEX

To lift clue-based query processing efficiency, this section introduces Roll-out-star R-tree (RSR-tree).

6.1 Roll-out-star

Fig. 5 (a) shows the "star" representation of spatio-textual context of POI $o \in D$. For every nearby o_i around o, the distance from o to o_i and the direction of edge (o, o_i) relative to the East are represented. In particular, the direction starts from 0^o degree for the East and going counter-clockwise through 90^o for North and 180^o for West and 270^o for South to 360^o degree for East again. **Fig.** 5 (b) shows that the "star" is "rolled out" into a 2-dimensional *distance-direction* space, called roll-out-star, with y-axis representing the distance away from o and x-axis representing the angle relative to the East. Note that the direction in **Fig.** 5 (b) is from 0^o to 720^o. But, only 0^o to 360^o need to be maintained in practice since 360^o to 720^o is the repeat of 0^o to 360^o. Sliding along x-axis from 0^o to 360^o is equivalent to rotating around o for 360 degrees.



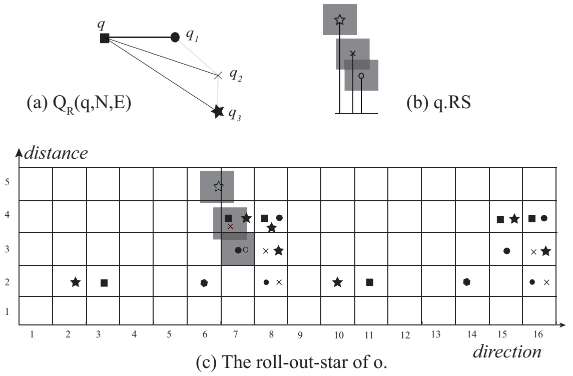
(a) Spatio-textual context of o

(b) Roll-out-start representation of spatial context of o

Figure 5: The roll-out-star of a POI $o \in D$.

The roll-out-star of every $o \in D$ is discretized using the same grid, i.e., uniformly partitioning the 2-dimensional *distance-direction* space. For example, the roll-out-star in **Fig.** 5 (b) is discretized into 16 × 5 cells as shown in **Fig.** 6 (c). Please note that each grid cell only records the categories of the POIs falling in the cell rather than the POIs themselves. A category is recorded once only even though multiple POIs are of this category. In the rest of this work, the term *roll-out-star* of o refers to the discretized roll-out-star, denoted as $o.RS$.

For the POIs $o_1, \cdots, o_l \in D$, we define the superimposition over their roll-out-stars $o_i.RS$, $1 \leq i \leq l$. In the superimposed roll-out-star, the category in each grid cell (a, b) is the union of the categories in the same grid cells of the original roll-out-stars. Suppose a pseudo POI o' whose roll-out-star is the superimposed

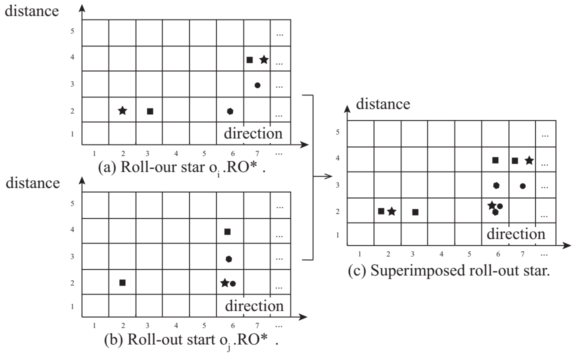


(a) Q_R(q,N,E)

(b) q.RS

(c) The roll-out-star of o.

Figure 6: Estimate spatio-textual context similarity using roll-out-star.



(a) Roll-our star o_i.RO* .

(b) Roll-out start o_j.RO* .

(c) Superimposed roll-out star.

Figure 7: Superimposition of roll-out-stars.

roll-out-star. Formally, for every grid cell (a, b),

$$o'.RS(a, b) := \cup_{1 \leq i \leq l}(o_i.RS(a, b)). \quad (5)$$

where $RS(a, b)$ is the set of categories in grid cell (a, b) of roll-out-star. **Fig.** 7 shows an example where two roll-out-stars in (a) and (b) are superimposed into the one in (c).

6.2 Similarity Estimation

Let $gc(q_i)$ be a grid cell with category $q_i.cid$. Given a query $Q_R(q, N, E)$ and a POI $o \in D_R(q.cid)$, the spatio-textual context similarity between o and q can be estimated using $o.RS$. **Fig.** 6 shows an example. First, the query in (a) is represented in the form of roll-out-star, denoted as $q.RS$, as shown in (b). Then, a grey region for each clue POI $q_i \in N \setminus \{q\}$ is generated, denoted as $gr(q_i)$. The grey region is in the same shape as a grid cell in $o.RS$. Once a clue POI $q_i \in N \setminus \{q\}$ is selected as q^m, $q.RS$ slides from 0^o to 360^o along the *direction* dimension of $o.RS$ (i.e., rotation), and moves in *distance* dimension of $o.RS$ (i.e., scaling) such that $gr(q^m)$ exactly matches one grid cell $gc(q^m)$. As shown in **Fig.** 6 (c), suppose q^m is q_1. $gr(q_1)$ is moved to exactly match grid cell (7,3) which is with category $q_1.cid$. Meanwhile, the grey region of each other clue POI $q_i \in N \setminus \{q, q^m\}$, e.g., $gr(q_2)$ and $gr(q_3)$ in **Fig.** 6 (c), is moved to the new location accordingly. The query is in the form $N_G = \{q, gr(q^m), gr(q_1), \cdots, gr(q_n)\}$. In the next step, the matching instance of N_G, denoted as Λ_G, is identified. In particular, $\Lambda_G = \{o, gc^m_{max}, gc_{1max} \cdots, gc_{nmax}\}$ where o matches q, gc^m_{max} matches $gr(q^m)$ and gc_{imax} matches $gr(q_i)$ for $1 \leq i \leq n$. Similar to Section 4, gc_{imax} for $1 \leq i \leq n$ is the closest grid cell, with category $q_i.cid$, to $gr(q_i)$ in terms of Euclidean distance in the original space (see Appendix-C [18]). It is allowed that some items in Λ_G are identical (i.e., $gc_{imax} = gc_{jmax}$, $i \neq j$, $1 \leq i, j \leq n$).

Note that we only match the grey region of q^m with the cell from 0^o to 360^o. Suppose a grid cell close to 360^o in $o.RS$ is with category $q^m.cid$. By moving the grey region of q^m to match this cell, other clue POIs in the query are moved to their new locations accordingly. Given the grey region of each clue POI (e.g., q_3 in **Fig.** 6 (c)), computing the Euclidean distance to each cell in $\pm 180^o$ with the same category is performed in the original space. Such a cell may be with degree less than 360^o (e.g., the cell covering o_3) or greater than 360^o (e.g., the cell covering o_{13}). To facilitate the computation, the direction dimension of roll-out-star is from 0^o to 720^o instead of from 0^o to 360^o.

Using $o.RS$, the spatio-textual context similarity between q and o is estimated as follows.

$$SCsim^*(q, o) = \max_{q^m \in N \setminus \{q\}, gc(q^m) \in o.RS} SCsim^*(q, o, gr(q^m), gc(q^m)). \quad (6)$$

where

$$SCsim^*(q, o, gr(q^m), gc(q^m)) = \gamma^* \max_{a_i \in N_G, b_i \in \Lambda_G} \tau_i * S(d_E(a_i, b_i)).$$

$$\gamma^* = S(\min(T5, T6)).$$

$$T5 = |gr(q^m).dist_{max} - gc(o^m).dist_{min}|.$$
$$T6 = |gr(q^m).dist_{min} - gc(o^m).dist_{max}|.$$

where a_i in N_G matches b_i in Λ_G, $d_E(a_i, b_i)$ is the Euclidean distance in the original space between a_i and b_i (see Appendix-C [18]), $gr(q^m).dist_{min}$ and $gr(q^m).dist_{max}$ are the minimum and maximum distance of $gr(q^m)$ to q in distance dimension in $q.RS$, $gc(o^m).dist_{min}$ and $gc(o^m).dist_{max}$ are the minimum and maximum distance of $gc(o^m)$ to o in distance dimension in $o.RS$.

LEMMA 2.

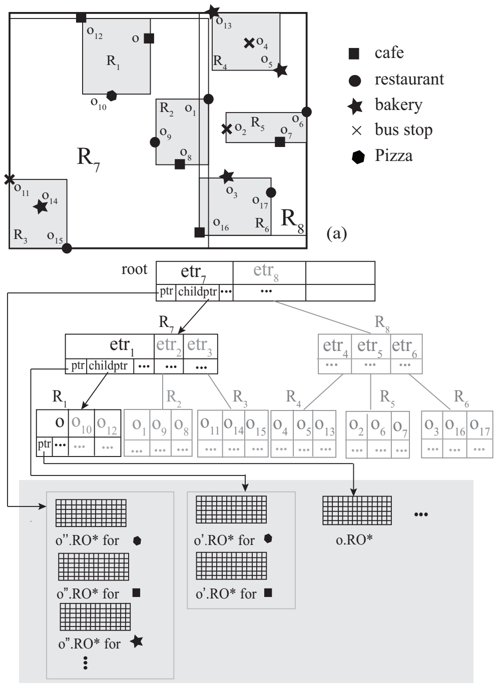
$$SCsim^*(q, o) \geq SCsim(q, o). \quad (7)$$

PROOF. Given o, suppose I is the matching instance with the highest $SCsim(q, o)$. Suppose $q_i \in N$ matches $o_i \in I$, the Euclidean distance between q_i and o_i, $d_E(q_i, o_i)$, is used in $S(d_E(q_i, o_i))$. For the grey region of q_i, $gr(q_i)$, suppose the closest grid cell, with category $q_i.cid$, is $gc(q_i)$. $d_{min}(gr(q_i), gc(q_i))$ is used in the sigmoid function. It is true that $d_{min}(gr(q_i), gc(q_i))$ is not greater than $d_E(q_i, o_i)$. So, $S(d_{min}(gr(q_i), gc(q_i))) \leq S(d_E(q_i, o_i))$. In addition, γ^* is not less than γ. As a result, we conclude $SCsim^*(q, o) \geq SCsim(q, o)$. □

LEMMA 3. *Given $Q_R(q, N, E)$ and a pseudo POI o' whose roll-out-star is the superimposed roll-out-stars of a number POIs, $o_1, \cdots, o_l$, we have*

$$SCsim^*(q, o') \geq SCsim^*(q, o_i),\ 1 \leq i \leq l. \quad (8)$$

PROOF. In each $o_i.RS$ for $1 \leq i \leq l$, $SCsim^*(q, o_i)$ is the maximum $SCsim^*(q, o_i, gr(q^m), gc(q^m))$ for all q^m and o^m. In $o'.RS$, $SCsim^*(q, o')$ is the maximum $SCsim^*(q, o', gr(q^m), gc(q^m))$ for all q^m and o^m. Given particular q^m and o^m, it must be true $SCsim^*(q, o', gr(q^m), gc(q^m))$ is not less than $SCsim^*(q, o_i, gr(q^m), gc(q^m))$ for $1 \leq i \leq l$. So, $SCsim^*(q, o') \geq SCsim^*(q, o_i)$ for $1 \leq i \leq l$. □



(b) RSR-tree.

Figure 8: Roll-out star R-tree.

6.3 Roll-out-star R-Tree

Roll-out-star R-Tree (RSR-tree) is an extended R-tree which supports clue-based spatio-textual query processing. First, an R-tree [21] (or R*-tree [22]) is constructed over the POIs in D. Then, spatio-textual context information is inserted into each node. For a leaf-node, each entry o contains $(o.id, o.cid, o.loc, o.ptr)$ where $o.id$, $o.cid$ and $o.loc$ are the identity, the category identify and the location of POI o, and $o.ptr$ refers to the roll-out-star of POI o. In a non-leaf node, each entry etr contains $(etr.MBR, etr.cids, etr.childptr, etr.ptr)$ where $etr.MBR$ is the minimum bound rectangle of POIs under the entry, $etr.cids$ is the list of categories of the POIs under the entry, $etr.childptr$ refers to a child node and $etr.ptr$ refers to the superimposed roll-out-stars of all POIs under the entry; note that, for each category in the entry, a superimposed roll-out-star is maintained. All roll-out-stars of the RSR-tree are stored in an external file. **Fig.** 8 presents an RSR-tree and the external file where $o'.RS$ for *café* is the superimposition of roll-out-stars of o, o_{12} and $o''.RS$ for *café* is the superimposition of roll-out-stars of o, o_{12}, o_8.

Given a POI dataset D, the space requirement for maintaining the roll-out-stars is $\sum_{cid} |B_{cid}| \log |B_{cid}|$ where $B_{cid} = \theta\beta\pi\xi^2$. In particular, ξ is the the maximum value in distance dimension of roll-out-star, and $\pi\xi^2$ is the region covered by a roll-out-star, θ is the number of POIs per unit region on average and β is the number of bits used to represent a category id. One method to reduce storage requirement is to compress the roll-out-stars near the leaf where most grid cells are empty. That is, we only record the grid cells which are unempty.

Suppose o is the truly matching POI of the querying POI q and o_i is the truly matching POI of some clue POI q_i. If a user believes that the distance between the querying POI q and some clue POI q_i may be greater than ξ, the query cannot be solved using the RSR-tree. In this situation, the algorithm 1 will be used. While ξ should be large enough to support every query, it is not plausible to set ξ too large. This is because the spatio-textual context is regarding the circumstances in proximity and the large ξ leads to large roll-out-star, i.e., greater storage requirement. In this work, $\xi = 5km$ by default.

If the POI database is updated, the R-tree and spatio-textual context of some POI may be updated correspondingly. If a POI changes its category or a POI is inserted/deleted, for any POI whose spatio-textual context is affected, the roll-out-star referred by the leaf node is updated first and then the update is propagated upwards to the roll-our star referred by the non-leaf node. The number of roll-out-star updated is $\theta\pi\xi^2$. The R-tree is updated, as described in [21], only if a POI is inserted/deleted.

6.4 Query Processing with RSR-tree

Given a clue-based spatio-textual query $Q_R(q, N, E)$, the RSR-tree is browsed following the best-first strategy. The pseudo code is illustrated in Algorithm 3. First, the root is visited. For each entry etr, if $etr.MBR$ overlaps with R and $etr.cids$ includes the querying category, this entry is inserted into a heap H, which is initially empty. For each entry $etr \in H$, the roll-out-star for the querying category, say RS, is loaded into memory from the external file. Let o_{etr} be the pseudo POI whose roll-out-star is RS. $SCsim^*(q, o_{etr})$ is computed using Equation (6) as the credit of etr. Then, the entry $etr \in H$ with the highest credit is processed by visiting the child node referred by $etr.childptr$; and etr is replaced in H by those entries in the child node, each of which has MBR overlapped with R and has $cids$ including the querying category. This operation is repeated until the entry in H with the highest credit is an entry of a leaf node. Suppose the POI maintained in this entry is o. The spatio-textual context similarity between q and o, $SCsim(q, o)$, is computed using Equation (3) and the value is set to cur. Any entry in H is deleted if its credit is less than cur according to Lemma 2 and Lemma 3. This operation continues and cur is updated if the higher spatio-textual context similarity is found. Once H is empty, the POI associated with cur is the solution of the query. This algorithm can be directly extended to searching for k POIs in $D_R(q.cid)$ with the highest spatio-textual context similarities.

7. ENSEMBLE METHOD

Given a clue-based spatio-textual query $Q_R(q, N, E)$, the ensemble method generates a number of new queries based on the original query $Q_R(q, N, E)$. Specifically, for each clue POI $q_i \in N \backslash \{q\}$, we generate a new query $Q_R(q_i, N, E)$ where q_i is treated as the querying POI, the original query POI q is treated as a clue POI. Each query (including the original and generated) retrieves ζ matching instances with the highest spatio-textual context similarities, and the ζ matching instances are sorted in descending order in a list l_i. So, we have $|N|$ sorted lists, denoted as L. For a matching instance I in the sorted lists, its score is

$$Score(I) = \sum_{l_i \in L} pos(I, l_i). \qquad (9)$$

where $pos(I, l_i)$ is the position of I in the sorted list l_i. The k matching instances with the highest scores are identified. From each of the k identified matching instances, the POI matching the original querying POI q is extracted and returned. An example is shown in **Fig.** 9 where an ensemble method is applied. The $\zeta = 5$ matching instances with the highest spatio-textual context similarities are identified for $Q_R(q, N, E)$, $Q_R(q_1, N, E)$, $Q_R(q_2, N, E)$ and $Q_R(q_3, N, E)$ respectively. I_1, I_4, I_3 are the 3 matching instances with the highest scores. The ensemble method serves two purposes.

Algorithm 3: Query processing with RSR-tree

```
Input: $Q_R(q, N, E)$, RSR-tree.
Output: The solution of $Q_R(q, N, E)$.
$H \leftarrow \emptyset$ ;
for each entry etr in RSR-tree root with overlap with $R$ do
    if etr has roll-out-star for the querying category then
        $etr.credit \leftarrow SCsim^*(q, o_{etr})$;
        $H \leftarrow etr$;
while $H$ is not empty do
    $etr \leftarrow$ entry in $H$ with highest $credit$;
    if etr is an entry in a non-leaf node then
        $ChildNode \leftarrow$ the node referred by $etr.childptr$;
        for each entry $etr'$ in $ChildNode$ do
            if etr has roll-out-star for the querying category then
                $etr'.credit \leftarrow SCsim^*(q, o_{etr'})$;
                $H \leftarrow etr'$;
        $etr$ is deleted from $H$;
    else
        // $etr$ is an entry of a leaf node
        $o \leftarrow$ the POI maintained in $etr$;
        $cur \leftarrow SCsim(q, o)$;
        $o_{cur} \leftarrow o.id$;
        for each entry $etr'$ in $H$ do
            if $etr'.credit < cur$ then
                $etr'$ is deleted from $H$;
return $o_{cur}$;
```

- First is to reach more reliable solution through cross-validation. That is, it is very likely that a matching instance I truly matches N if I has been retrieved for multiple times using ensemble method.
- Second, the truly matching POI of the original querying POI q may be absent in the database due to data quality problems. It is highly challenging since it is impossible to know this situation in advance. The ensemble method is an effective approach to attack this problem. Even though the truly matching POI of q is absent in the database, we still have chances to retrieve the relevant matching instances and infer the approximate location of the original querying POI because of the spatial relationships provided in the query.

8. EXPERIMENTS

This section reports test results. We are interested in: **i**) the accuracy of proposed spatio-textual similarity metric and the impact of various factors to the accuracy; **ii**) the query processing efficiency and the effectiveness of RSR-tree. All algorithms have been implemented using C++ and performed on a machine with Intel(R) Core(TM) i5-4310U CPU @ 2.00GHz 2.60GHz, 8GB RAM and Window 7 Enterprise operating system.

8.1 Data sets

We use the real POI data sets in Beijing, one of the largest cities in the world. The data sets are purchased from datatang.com which provides high quality data for various research purposes. The POI data sets are of four years i.e., 2005, 2007, 2009 and 2013. **Fig.** 10 illustrates the POIs in the central region of Beijing in the four data sets. The major categories of the POIs are illustrated in Table 2. The POI data sets present the following trends: (a) the number of POIs in each category increases gradually from 2005 to 2013; (b) the category of POIs changes over time, for example, about 5 percent of POIs in 2009 have the category changed in 2013. The POI data sets in 2005, 2007, 2009 and 2013 are used to represent four POI databases of different data qualities. The quality increases from 2005 to 2013.

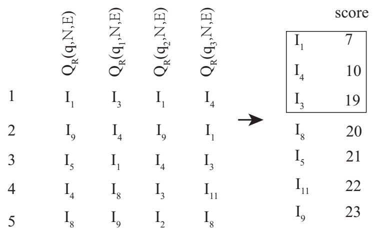

$Q_R(q,N,E)$
$Q_R(q_1,N,E)$
$Q_R(q_2,N,E)$
$Q_R(q_3,N,E)$
score
1 I_1 I_3 I_1 I_4
2 I_9 I_4 I_9 I_1
3 I_5 I_1 I_4 I_3
4 I_4 I_8 I_3 I_{11}
5 I_8 I_9 I_2 I_8
I_1 7
I_4 10
I_3 19
I_8 20
I_5 21
I_{11} 22
I_9 23

Figure 9: Ensemble method.

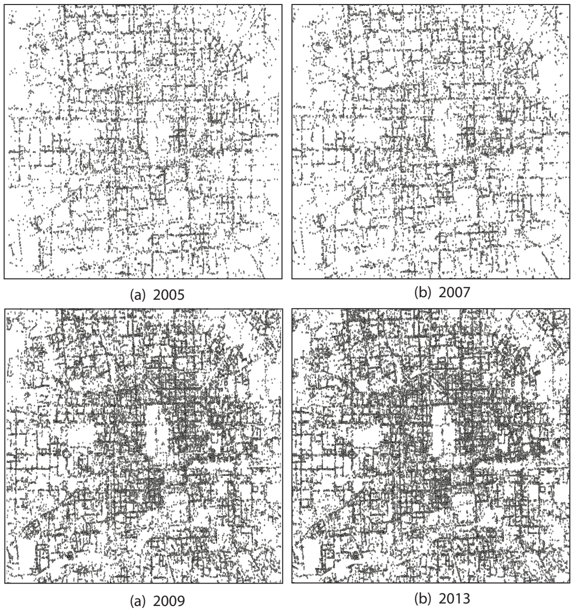


Figure 10: POIs in the central area of Beijing.

The clue-based spatial queries are generated using the 2013 data set. At the same test setting, 100 clue-based spatio-textual queries are generated and the averaged results are reported. Specifically, 100 POIs are selected, as the querying POIs, from the POIs which are present in all four data sets. For each querying POI q, a number of clue POIs are randomly selected from the 2013 data set in the proximity around q. By default, the query region R is the entire Beijing city and the number of clue POIs (excluding the querying POI) is 4. To mimic the approximate nature of human in estimation of distance and relative direction, the clue POIs are shifted to new locations by following the bivariate normal distribution with probability density function normal distribution.

$$p(x, y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp^{[-\frac{z}{2(1-\rho^2)}]} \tag{10}$$

Table 2: Major Categories of POIs

Name	2005	2007	2009	2013
Total	34159	36511	118643	182323
Market	3821	4956	19831	24705
Restaurant	2092	4123	14208	20001
Company	3125	3589	13082	15509
Bank	1629	1868	4256	7269
Public Toilet	1872	2149	5726	6903
Residential District	1890	2539	5290	6446
School	2193	2458	4001	5893
Car Park	671	1382	3876	4957
Government Building	1200	1892	3100	4675
Pharmacy	892	1287	3161	3759
Sport Club	623	1202	2605	3139
Factory	1586	1890	2248	2613
Business building	569	940	1693	2022
Bridge	998	1106	1264	1572
Gasoline Station	266	287	345	1029

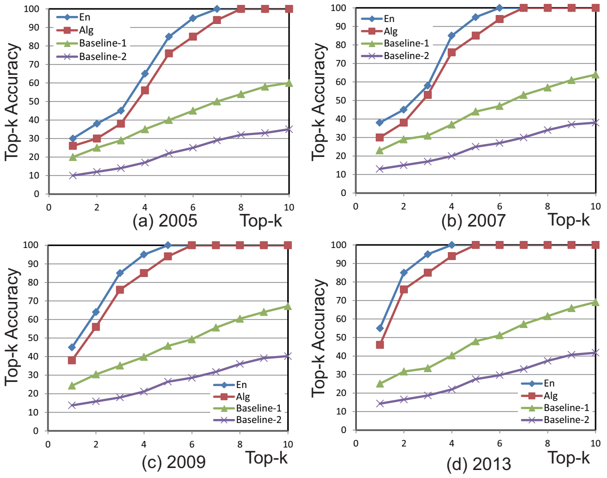


Figure 11: The impact of k to accuracy.

where

$$\begin{aligned} z &\equiv \frac{(x-\mu_x)^2}{\sigma_x^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y} + \frac{(y-\mu_y)^2}{\sigma_y^2} \\ \rho &\equiv cov(x,y) \end{aligned}$$

ρ is the correlation of x and y. Generating data pairs (x, y) from a bivariate normal distribution with a specific correlation ρ, $X \sim N(\mu_x, \sigma_x^2)$, and $Y \sim N(\mu_y, \sigma_y^2)$ [23]. Given a clue POI $q_i(x_{q_i}, y_{q_i})$, we have $\mu_x = x_{q_i}$, $\mu_y = y_{q_i}$, $\rho = 0$, and $\sigma_y \equiv \sigma_x$. Clearly, the deviation of a clue POI from its true position only depends σ_x and σ_y.

8.2 Accuracy

This section reports the proposed query processing methods in terms of top-k accuracy. Top-k accuracy is widely used to measure how often the correct answer is found within the top-k answers reported by an algorithm (e.g., [24]). In specific, the top-k POIs of a query are the k POIs with the highest spatio-textual context similarities. If the truly matching POI can be found in the top-k POIs, $\varepsilon = 1$; otherwise, $\varepsilon = 0$. Given η queries of the same setting, the top-k accuracy is defined as $\frac{1}{\eta}\sum_{i=1}^{\eta}\varepsilon_i$, that is, the ratio of queries where the truly matching POI can be found in the top-k POIs. By default, $k = 5$ and $\eta = 100$ in the tests.

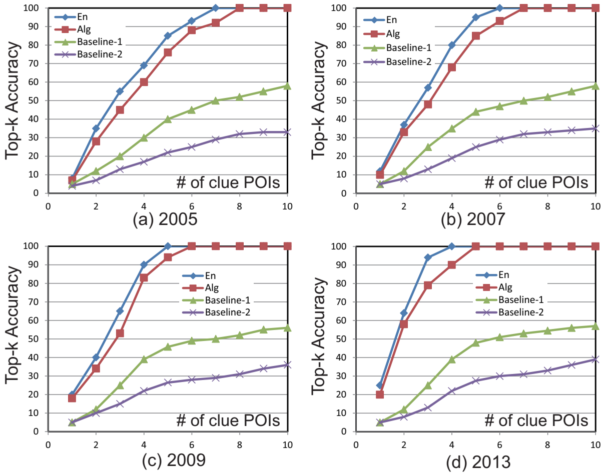


Figure 12: The impact of the number of clue POIs to accuracy.

We compare the top-k accuracies of the proposed algorithm, the proposed ensemble method, and two baseline algorithms, denoted as *Alg*, *En*, *Baseline-1* and *Baseline-2* respectively. While the procedure is same as *Alg*, the baseline algorithms apply different spatio-textual context similarity metrics.

- In Baseline-1, the spatio-textual context similarity is measured by the distance only and the directional information is ignored. Formally,

$$SCsim_{bl1}(q, o) = \sum_{q_i \in N} S(d_E(q_i, o_i)). \qquad (11)$$

Given the querying POI q and a POI $o \in D_R(q.cid)$, for each clue POI $q_i \in N \setminus \{q\}$, the matching POI of q_i is $o_i \in D_R(q_i.cid)$ if the difference between $d_E(o, o_i)$ and $d_E(q, q_i)$ is minimum compared to any other POI $o_i' \in D_R(q_i.cid)$ in the data set. If more than one clue POIs with the same category have the same matching POI in the data set, the Augmenting Path algorithm is applied as discussed in Section 5 to make sure one to one match for this category. The aim of Baseline-1 is to evaluate the value of directional information to the query accuracy.

- In Baseline-2, the spatio-textual context similarity considers both directional information and distance. But the distance is directly applied, i.e., the sigmoid function is not applied to normalize the distance. Formally,

$$SCsim_{bl2}(q, o) = \qquad (12)$$

$$\min_{q^m \in N, o^m \in D_R(q^m.cid)} SCsim_{bl2}(q, o, q^m, o^m).$$

$$SCsim_{bl2}(q, o, q^m, o^m) =$$

$$\gamma \min_{I \subseteq D_R} \sum_{q_i \in N, o_i \in I} \tau_i * d_E(q_i, o_i).$$

The smaller $SCsim_{bl2}(q, o)$ is, the higher the spatio-textual context similarity between q and o is. The aim is to evaluate the effectiveness of sigmoid function to handle the data quality problems.

Impact of k: In **Fig.** 11, x-axis indicates k and y-axis is the top-k accuracy. When k changes from 1 to 10, the top k accuracy steadily increases for all methods on all data sets. The superiority of *Alg* and *En* is significant. Compared to *Alg*, *En* always shows the enhanced accuracy. This is consistent with the analysis in Section 7.

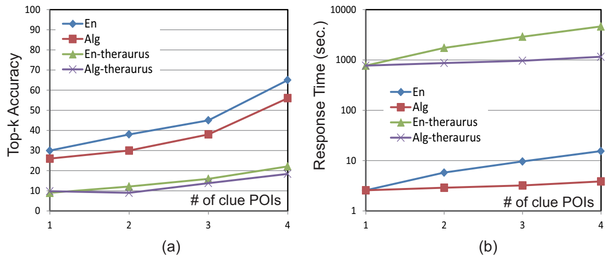


Figure 13: The impact of thesaurus to accuracy and query response time.

The accuracy of Baseline-1 is the lowest at all settings for all data sets. This result verifies the value of directional information to the query accuracy. The accuracy of Baseline-2 is better than that of Baseline-1 but is much lower than that of Alg. To understand the reason behind, we use the example in **Fig.** 3 (b) where q and o are truly matched. If o_2 is missing or its category is updated, q_2 has to match with o_4. The long distance between q_2 and o_4 dominates the spatio-textual context similarity in Baseline-2. That is, a single data quality problem has significant impact to the spatio-textual context similarity. In contrast, our spatio-textual context similarity metric strategically solves this issue by trivializing the impact of single data quality problems.

Impact of clue POI number: More clue POIs in a query (excluding the querying POI) mean more spatio-textual context information is provided and thus lead to the higher accuracy. This is verified by the case study shown in Appendix-B [18]. **Fig.** 12 shows, when the number of clue POIs in a query increases from 1 to 10, the accuracy of a query quickly increases. Compared to the baseline algorithms, the superiority of Alg and EN is significant.

Impact of thesaurus: For each POI in the query, the category is extended to include thesaurus in order to handle the inexact category issue in a query. Once the thesaurus is included, it is unsurprising that the accuracy decreases for all methods on all data sets as shown in **Fig.** 13 (a). In **Fig.** 13 (b), the query processing efficiency with thesaurus is compared with that without thesaurus. The inclusion of thesaurus increases the search space which requires much more processing time. So, we recommend that the thesaurus is applied only when the output without thesaurus has been rejected by user.

Impact of β: The spatio-textual context similarity $SCsim(q, o, q^m, o^m)$ defined in Equation (2) applies a sigmoid function where the parameter β decides the slope of curve (see **Fig.** 4 (b)). The setting of β provides user the capability to adapt the query to different scenarios. For example, suppose q_i matches o_i. If a user believes that a distance more than 500 meters between q_i and o_i is because of data quality problem, $\beta = 0.01$ should be set, i.e., q_i and o_i is ignored in the similarity metric; if a user believes that a distance more than 50 meters between q_i and o_i is because of data quality problems such as in the landscape design application described in Section 1, $\beta = 0.1$ should be set. The test results shown in Figure 14 indicates that $\beta = 0.01$ is the best setting in the context of POI search in a city.

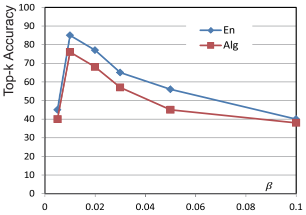


Figure 14: The impact of β.

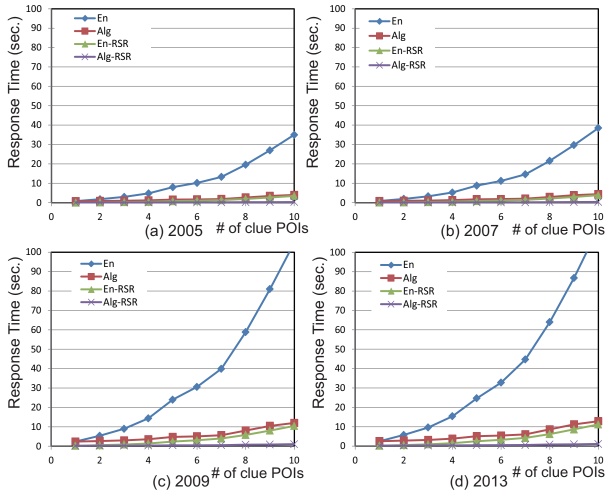


Figure 15: The effectiveness of RSR-tree.

8.3 Efficiency and RSR-tree

By default, the roll-out-star applied in RSR-tree is $< 25, 15 >$, i.e., 25 meters and 15^o in distance and direction dimension respectively for each grid cell; the maximum value in distance dimension is $5km$ by default. In **Fig.** 15, the query processing efficiency with RSR-tree is compared with that without RSR-tree. When the number of clue POIs in a query (excluding the querying POI) increases from 1 to 10, the processing time with RSR-tree dramatically reduced by up to 10 times.

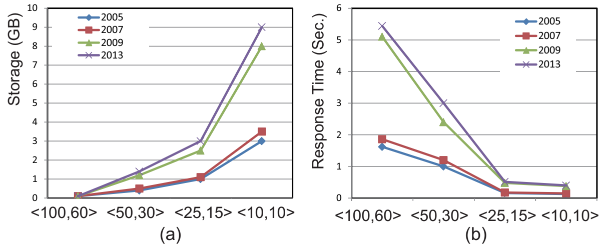


Figure 16: Impact of gird granularity to RSR-tree.

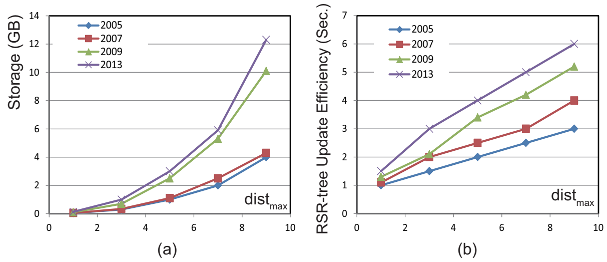


Figure 17: Impact of ξ to RSR-tree.

For every roll-out-star in the RSR-tree, the *distance-direction* space is partitioned using the same grid. At different partition setting from $< 100, 45 >$ to $< 5, 10 >$, the storage requirement and the filtering capability of RSR-tree is presented in **Fig.** 16. From $< 100, 45 >$ to $< 5, 10 >$, the number of grid cells increases by more than 800 times and the size of file maintaining the roll-out-stars increase much slower as shown in **Fig.** 16 (a). This is because (i) empty grid cells of roll-out-star are not maintained, (ii) the number of empty grid cells increases from coarse grid to finer grid, and (iii) a cell in coarse grid tends to have more categories than that in finer grid. Since the query region for all queries is the entire city, the filtering capability of RSR-tree of different roll-out-star settings can be evaluated by the query processing time as shown in **Fig.** 16 (b). The RSR-tree with finer grid setting has better filtering capability because the spatio-textual similarity estimation tends to be closer to the actual spatio-textual similarity.

As discussed in Section 6.3, the maximum value in distance dimension of roll-out-star, ξ, determines the query to be supported by RSR-tree. Given a query $Q_R(q, N, E)$, if a user believes that the distance between the querying POI q and a clue POI q_i is greater than ξ, the query cannot be solved using the RSR-tree. In this situation, Algorithm 1 will be used. The storage required for maintaining roll-out-stars at different settings of ξ is presented in **Fig.** 17(a). Once a POI has category changed, the time for updating the roll-out-stars of RSR-tree is shown in **Fig.** 17(b). The update is efficient since the number of roll-out-stars to be updated is localised.

9. CONCLUSIONS AND FUTURE WORK

The clue-based spatio-textual query explores the spatio-textual context as useful clue in the situation where the information is not enough to pinpoint the POI of interest. Besides the POI retrieval, this query can be found in an application where the space layout of objects is essential such as in landscape design and tactic analysis in Football game. The accuracy and efficiency of the proposed query processing methods in this work have been verified through extensive tests upon data sets of the real world. In the future, we plan to extend the proposed techniques of this paper, including the context similarity metrics and RSR-tree, from 2 dimensional space to 3 dimensional space. As a consequence, the clue-based spatio-textual query can be applied in applications where spatio-textual context similarity search in 3D space is essential, for example, search for 3D CAD design with the similar distribution of components and search for proteins with similar 3D structure.

10. ACKNOWLEDGEMENTS

This work is partially supported by Australian Research Council (DP160102114, LP130100164), Natural Science Foundation of China (61070024, 61433008, 61272179, 61232006), the MOE-China Mobile Fund (MCM20125021) and the Obel Family Foundation.

References

[1] P. Bouros, S. Ge, and N. Mamoulis. "Top-k Spatio-Textual Similarity Join". In: *PVLDB* 6.1 (2012), pp. 1–12.

[2] J. Rao, J. Lin, and H. Samet. "Partitioning strategies for spatial-textual similarity join". In: *Proc. of SIGSPATIAL workshop on BISIGSPATIAL*. 2012, pp. 824–835.

[3] H. Hu et al. "Top-k Spatio-Textual Similarity Join". In: *IEEE Trans. Knowledge and Data Engineering (TKDE)* 28.2 (2015), pp. 551–565.

[4] G. Cong, C. S. Jensen, and D. Wu. "Efficient retrieval of the top-k most relevant spatial web objects". In: *Proc. VLDB Endow.* 2.1 (2009), pp. 337–348.

[5] Z. Li et al. "IR-Tree: An Efficient Index for Geographic Document Search". In: *IEEE Trans. Knowledge and Data Engineering (TKDE)* 23.4 (2011), pp. 585–599.

[6] J. Fan et al. "SEAL: spatio-textual similarity search". In: *Proc. of VLDB*. 2012, pp. 824–835.

[7] I. D. Felipe, V. Hristidis, and N. Rishe. "Keyword Search on Spatial Databases". In: *Proc. of ICDE*. 2008, pp. 656–665.

[8] K. Deng et al. "Best Keyword Cover Search". In: *IEEE Trans. Knowledge and Data Engineering (TKDE)* 27.1 (2015), pp. 61–73.

[9] Y. Manolopoulos, A. Nanopoulos, and E. Tousidou. *Advanced Signature Indexing for Multimedia and Web Applications*. Reading, Massachusetts: Springer Science & Business Media, 2012.

[10] K. A.Nedas and M. J. Egenhofer. "Spatial-Scene Similarity Queries". In: *Transactions in GIS* 12.6 (2008).

[11] M. J. Egenhofer. "Query Processing in Spatial-Query-by-Sketch". In: *Journal of Visual Languages and Computing* 8.4 (1997), pp. 403–424.

[12] T. Brinkhoff, H. P. Kriegel, and B. Seeger. "Efficient processing of spatial joins using R-trees". In: *Proc. of ACM SIGMOD*. 1993, pp. 237–246.

[13] G. R. Hjaltason and H. Samet. "Incremental Distance Join Algorithms for Spatial Databases". In: *Proc. of ACM SIGMOD*. 1998, pp. 237–248.

[14] N. Mamoulis and D. Papadias. "Multiway spatial joins". In: *ACM transactions on database systems* 26.4 (2001), pp. 424–475.

[15] R. Cheng, D. Kalashnikov, and S. Prabhakar. "Evaluating probabilistic queries over imprecise data". In: *Proc. of ACM SIGMOD*. 2003, 551562.

[16] Y. Tao et al. "Indexing multi-dimensional uncertain data with arbitrary probability density functions". In: *Proc. of VLDB*. 2005, pp. 922–933.

[17] K. Zheng, P.-C. Fung, and X. Zhou. "K-Nearest Neighbor Search for Fuzzy Objects". In: *Proc. of ACM SIGMOD*. 2010.

[18] *Appendix*. URL: https://github.com/uqkdeng/Research-Repository/blob/master/Clue/Clue-appendix.pdf.

[19] C. Brewster and Y. Wilks. "Ontologies, taxonomies, thesauri learning from texts". In: *Proceedings of the Use of Computational Linguistics in the Extraction of Keyword Information from Digital Library Content*. 2004.

[20] D. B. West. *Chapter 3, Introduction to Graph Theory (2nd ed.)* Prentice Hall, 1999.

[21] A. Guttman. "R-Trees: A Dynamic Index Structure for Spatial Searching". In: *Proc. of ACM SIGMOD*. 1984, pp. 47–57.

[22] N. Beckmann et al. "R*-tree: an efficient and robust access method for points and rectangles". In: *Proc. of ACM SIGMOD*. 1990, pp. 322–331.

[23] *Generating Data with a Specified Correlation*. URL: http://www.uvm.edu/~dhowell/StatPages/More_Stuff/CorrGen.html.

[24] A. Penta et al. "Discovering cross-language links in Wikipedia through semantic relatedness". In: *proceedings of the 20th European Conference on Artificial Intelligence*. 2012.