计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20140686
52(9):1954-1964,2015

一种语义增强的空间关键词搜索方法

韩 军 范 举 周立柱
(清华大学计算机科学与技术系 北京 100084)
(hanj04@gmail.com)

Semantic-Enhanced Spatial Keyword Search

Han Jun, Fan Ju, and Zhou Lizhu
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract Spatial keyword search finds the points-of-interest (POIs) which are not only relevant to users' query intent, but also close to query location. Spatial keyword search has many important applications, such as map search. Previous methods for spatial keyword search have the limitation that they only consider textual relevance of POIs to query keywords, and neglect the semantics of queries. So these methods may not be able to return relevant results or return many irrelevant results. To address this problem, this paper introduces a semantic-enhanced spatial keyword search method, named S^3 (semantic-enhanced spatial keyword search). Given a query, S^3 analyzes the semantics of the query keywords to measure semantic distances of POIs to the query. Then, it utilizes a novel POI ranking mechanism by combining both semantic and spatial distance for effective POI search. S^3 has the following challenges. Firstly, S^3 introduces knowledge bases to help capture query semantics and introduces a ranking scoring function that considers both semantic distance and spatial distance. Secondly, it calls for instant search on large-scale POI data sets. To address this challenge, we devise a novel index structure GRTree, and develop some effective pruning techniques based on this structure. The extensive experiments on a real dataset show that S^3 not only produces high-quality results, but also has good efficiency and scalability.

Key words spatial keyword search; semantic enhancement; knowledge base; semantic distance; instant search

摘 要 空间关键词搜索立足于查找满足用户查询意图且空间距离相近的兴趣点(point of interest,POI),在地图搜索等领域有着广泛的应用.传统的空间关键词搜索方法仅考虑关键词与POI点在文本上的匹配程度,忽略了查询的语义信息,因而会导致相关结果丢失以及无关结果引入等问题.针对传统方法的局限,提出了语义增强的空间关键词搜索方法 S^3(semantic-enhanced spatial keyword search).该方法对查询关键词中包含的语义信息进行分析,并结合语义相关性和空间距离对POI点进行有效的排序.S^3 方法主要有以下2个技术挑战:1)如何对语义信息进行分析.为此,S^3 引入了知识库对POI数据进行语义扩充,并提出了一种基于图的语义距离度量方式.结合语义距离和空间距离,S^3 给出POI点的综合排序方案.2)如何在大规模数据上即时地返回top-k搜索结果.针对这一挑战,提出了一种新型的语义-空间混合索引结构GRTree(graph rectangle tree),并研究了有效的剪枝策略.在大规模真实数据集上的实验表明,S^3 不仅能够返回更为相关的结果,而且有着很好的效率和可扩展性.

关键词 空间关键词搜索;语义增强;知识库;语义距离;即时搜索

中图法分类号 TP391

收稿日期:2014-07-25;**修回日期**:2014-11-20
基金项目:国家自然科学基金项目(61272090)

随着移动终端的迅猛发展和地图搜索需求的日益增长,空间关键词搜索得到了学术界和工业界的广泛重视.针对空间上的一组兴趣点(point of interest,POI),空间关键词搜索需要用户输入查询空间位置和一组关键词,之后返回与用户查询意图相关且距离相近的 top-k 个 POI 点作为结果.

然而,传统空间关键词搜索方法存在以下局限性:它们仅依据关键词和 POI 点在文本上的匹配程度来判断相关性,而忽略了查询包含的语义信息.这会导致以下 2 方面问题:

1) 这些方法可能会丢失很多与查询相关的结果,这在地图搜索等应用中表现得尤为突出.在这些应用中,关键词的匹配程度并不能准确地反映语义,因此即便与查询语义相关,POI 点也很可能因为与关键词没有匹配而无法得到返回.例如,当用户希望查找治疗中风(stroke)的医院时,她可能输入查询"Stroke Hospital",此时 POI 点"Uab Hospital"是一条相关结果,因为它的神经科可以治疗中风;然而如果它的文本描述中不包含"Stroke",则它可能不会进入到 top-k 结果中.

2) 由于仅考虑关键词匹配程序,所有文本描述中含"Hospital"的 POI 点都有可能被返回,然而它们之中只有少量为可以治疗中风的医院.这样一来用户不得不在大量无关的结果中筛选出真正相关的结果.

为了解决上述问题,本文提出语义增强的空间关键词搜索方法 S^3(semantic-enhanced spatial keyword search).针对用户查询,S^3 可以分析查询关键词包含的语义信息,并度量 POI 点与查询在语义上的距离;之后,S^3 综合考虑语义距离以及空间距离对 POI 点进行有效的排序,返回最相关 top-k 结果.以上文的查询"Stroke Hospital"为例,S^3 会分析出 POI 点"Uab Hospital"与"Stroke"在语义是相关的,并对相关程度进行量化,计算语义距离;之后,结合该点的空间距离,给出一个综合的排序分数.

本文主要研究 S^3 中的以下技术挑战:

1) 如何对语义距离进行度量,并结合空间距离给出综合的排序分数.针对该挑战,我们提出使用知识库(如 Freebase[①])对 POI 点进行语义扩充.由于知识库中包含大量实体、属性和实体关联,扩充后能极大地丰富 POI 的语义信息.S^3 提出采用空间语义图来表示扩充之后的结果.基于该图,S^3 给出了一种新型的 POI 点排序打分方案:该方案通过空间语义图的路径距离来衡量语义距离,并结合空间距离给出综合分数.

2) 如何提高查询处理效率,满足用户在大规模空间数据上的即时搜索需求.S^3 提出了一种新型的索引 GRTree(graph rectangle tree).该索引对空间语义图进行层次化地划分,并维护每一个划分中包含的 POI 的空间信息.在查询处理的过程中,通过 GRTree 索引,S^3 可以渐进地估计图划分与查询的语义距离和空间距离的界,并基于此将距离过远的图划分尽早地放弃,从而提高效率.

综上所述,本文的贡献点主要包括:

1) 研究了语义增强的空间关键词搜索,提出了一种有效的方法 S^3.该方法利用知识库对 POI 进行语义扩充,并给出了有效的语义-空间排序方案.

2) 提出一种新的索引结构 GRTree.该索引通过对空间语义图的层次化划分,能够有效地过滤掉距离过远的 POI,提高搜索效率.

3) 在大规模的真实数据上进行了实验.实验表明:①与传统空间关键词搜索相比,S^3 能够返回更为相关的结果,有着更好的搜索质量;②S^3 利用 GRTree 索引可以在 30ms 以内返回查询,满足了即时查询的需求.

1　S^3 方法

1.1　方法概览

图 1 给出了 S^3 的概览,该方法主要解决一组 POI 上的搜索问题.其中,每个 POI 包含 2 部分信息:空间坐标(即经纬度)以及文本描述.图 1 给出了一些 POI 示例:每个 POI 都包含空间坐标以及如"Uab Hospital","Broughton"等文本描述.

为了有效地提供语义增强的空间关键词搜索,S^3 分为离线和在线 2 个环节.

1) 离线环节:POI 集合的语义扩充.如引言所述,POI 点的文本描述包含的语义信息十分局限,因此,S^3 利用知识库对 POI 集合进行语义扩充.

知识库包含大量现实世界中的实体以及实体之间的关系.直观来讲,知识库可以看作一个图:顶点表示实体,边表示实体之间的关系.例如,在图 1 中,"Uab Hospital"表示一个医院实体,"Neurology"表示神经科实体,它们之间的边表示"Neurology"是"Uab Hospital"的一个科室.

① http://www.freebase.com

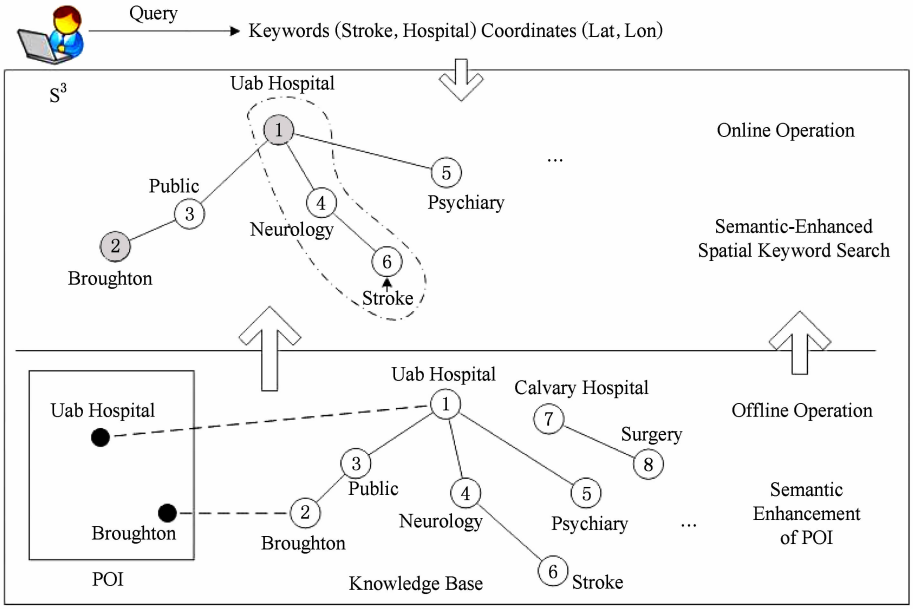


Fig. 1 Procedure of S³.

图 1 S³ 方法流程

基于知识库,本文按照以下方法对 POI 集合进行语义扩充:针对集合中的每个 POI,到知识库中寻找与之"匹配"的实体.具体方法是度量该 POI 点的名称对应的文本与知识库中实体名称的文本相似性①(如 Jaccard 相似性),并选择文本相似性最大的实体作为与该 POI 匹配到的实体.按照这样的方法,我们会得到知识库中所有与 POI 匹配的实体以及与这些实体连通的顶点,我们称之为空间语义图,如图 1 上半部分所示,其中灰色圆圈代表与 POI 点匹配到的顶点,称为空间顶点,如 Uab Hospital;白色圆圈表示与之连通的其他顶点,如 Stroke,Public 等.构建的空间语义图是语义增强空间关键词搜索的数据基础(圆圈内数字表示顶点编号).

2) 在线环节:语义增强的空间关键词搜索.基于构建好的空间语义图,S³ 回答用户提交的空间关键词查询.查询包含一个空间位置和若干关键词,如图 1 给出的示例"Stroke,Hospital".

回答用户查询的核心是在空间语义图上找出满足用户查询意图的空间顶点.为此,S³ 提出要度量空间顶点与查询的语义距离.在本文中我们使用关键词命中的顶点来代表关键词,从而将关键词与空间顶点之间的语义距离转化为关键词命中顶点与空间顶点的距离.以图 1 为例,关键词 Stroke 和 Hospital 分别命中顶点 v_1 和 v_6.S³ 通过度量这 2 个顶点与任意空间顶点 v_i 在图上的距离来衡量查询与 v_i 之间的语义距离,从而判断空间顶点 v_i 在语义上满足查询的程度:语义距离越近,满足的程度越高;反之越低.另一方面,S³ 要度量空间顶点与查询位置之间的空间距离:这个距离越近,说明用户越容易到达对应的 POI.

S³ 结合语义距离和空间距离对空间顶点进行排序,并将综合距离最小的前 k 个顶点作为结果返回.

1.2 语义增强空间关键词搜索的形式化定义

1) 空间语义图.本文使用 $G=(V_P \cup V_N, E)$ 表示空间语义图:V_P 表示知识库中与 POI 点匹配的顶点;V_N 表示其他与 V_P 连通的顶点;E 表示边集,其中每条边都包含一个权重,表示关系的紧密性.对于任意顶点 $v \in V_P$,令 $v.\rho$ 表示其空间位置.现有的研究给出了不同的权重赋值方案[1-2],S³ 可以用于处理任意的权重方案.在实验中,参考了文献[2]的方案,详见 4.1 节.图 2 下半部分给出了一个空间语义图的示例.其中,各个顶点的名称参见表 1;灰色圆圈(顶点 0~7)表示空间顶点,白色圆圈(顶点 8~17)表示其他顶点.

① POI 与知识库实体的匹配可以使用更为复杂的方式,如更有效的相似性函数或考虑知识库结构等.本文侧重于解决空间语义图上的搜索问题,不对匹配问题作过多的表述.

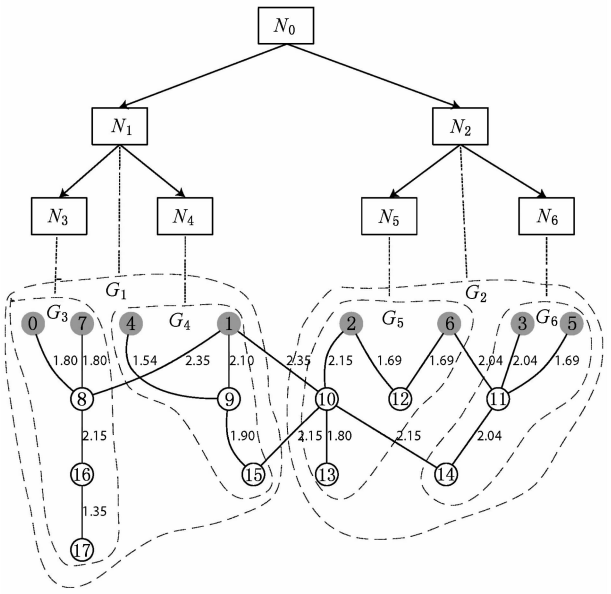


Fig. 2　Sample of GRTree and spatial semantic graph.

图 2　GRTRee 和空间语义图样例

Table 1　Name of Vertices

表 1　各顶点名称

v	Name	v	Name
0	Mempuhis Clinic	9	Psychiatry
1	Uab Hospital	10	Neurology
2	North Port Medical Center	11	Public Health
3	Broughton Hospital	12	Dentistry
4	Hemphill County Hospital	13	Stroke
5	Kingston Public Hospital	14	Wilson Disease
6	Gardon City Hospital	15	Dementia
7	Fox Chase Cancer Center	16	Heart Disease
8	Internal Medicine	17	Dry Cough

2) 查询. 本文研究空间关键词查询 $Q=\{\lambda,\rho,k\}$,其中 $\lambda=\{q_1,\cdots,q_{|Q|}\}$ 为关键词集合,ρ 为一个空间查询位置,k 为一个数字. 查询的结果是空间语义图中的一组空间顶点(每个空间顶点都对应一个 POI 点),按照下面排序模型对空间顶点进行排序,最终返回排序列表中前 k 个结果.

3) 排序模型. 排序模型包含语义距离和空间距离 2 部分,下面分别进行介绍:

① 语义距离衡量空间顶点 v 与关键词命中的顶点在空间语义图上的距离. 具体来说,考虑查询中的某一关键词 q,令 $q\in v$ 表示 q 命中顶点 v,即 q 被 v 的文本名称所包含[①]. 由于 q 可能命中多个顶点,本文采用与 v 在图上最近的顶点来度量 v 与 q 的语义距离,即 $Sem_G(q,v)=\min\limits_{q\in v_i,v_i\in V_P\cup V_N} dis_G(v,v_i)$. 其中,$dis_G(v,v_i)$ 表示顶点 v 与 v_i 在图 G 中最短路的(加权)距离. 在特殊情况下:如果 v 和 v_i 之间不存在路径,则认为语义距离 $dis_G(v,v_i)=\infty$;如果在 G 中,关键词 q 没有命中任何顶点,则同样认为 $Sem_G(q,v)=\infty$.

综合所有关键词,用户查询与空间顶点在 G 中的语义距离定义为 $Sem_G(Q,v)=\sum\limits_{q\in Q.\lambda} Sem_G(q,v)$.

② 空间距离衡量空间顶点 v 与查询位置之间的欧氏距离. 查询 Q 到空间顶点 $v\in V_P$ 之间的空间距离定义为 $Spa(Q,v)=dis(Q.\rho,v.\rho)$.

综合考虑语义距离和空间距离,本文使用以下公式对空间顶点进行排序:

$$D(Q,v)=\alpha\times\frac{Sem_G(Q,v)}{\max\limits_{v'\in V_P} Sem_G(Q,v')}+(1-\alpha)\times\frac{Spa(Q,v)}{\max\limits_{v'\in V_P} Spa(Q,v')},\tag{1}$$

其中 $\alpha\in[0,1]$ 为平衡 2 种权重的参数,可以通过实验进行调整;$\max\limits_{v'\in V_P} Sem_G(Q,v')$ 和 $\max\limits_{v'\in V_P} Spa(Q,v')$ 用于对语义距离和空间距离归一化.

基于式(1),定义语义增强的空间关键词搜索:

给定查询 $Q=\{\lambda,\rho,k\}$ 和空间语义图 $G=(V_P\cup V_N,E)$,结果集合 R 同时满足:1)集合 R 大小为 k,即 $|R|=k$;2)R 中元素均为空间顶点,即 $R\subseteq V_P$;3) $\forall v\in R,\forall u\in(V_P-R),D(Q,v)\leqslant D(Q,u)$.

2　GRTree

2.1　基本思想

S^3 最直观的搜索方法是首先将所有空间顶点按照空间距离进行排序,再根据语义距离进行筛选. 然而,这种分开索引的方法效率很低,因此我们考虑使用语义和空间的混合索引结构来提高效率. 一种最为常用的混合索引为 IRTree[3]. 该方法首先对空间进行层次划分,根据空间位置将空间顶点组织成一棵树,其中树的每个节点代表空间上的一个区域,父节点对应的区域包含子节点对应的区域,叶节点关联该区域内的 POI 点. 对每个树节点,预先计算节点包含的 POI 顶点到相关的所有关键词的语义距离,并维护在一个倒排索引中. 在查询处理的过程

① 关键词既可以命中空间顶点,也可以命中其他节点. 不需要对命中节点的类型进行区分,而是统一地度量距离.

中,针对每个节点计算空间距离和语义距离的界,并将距离过大的节点尽早地过滤掉.然而,在空间语义图中,由于空间顶点可能与全图的关键词均相关,于是所有包含该顶点的 IRTree 节点的倒排索引均要维护所有关键词的语义距离,从而导致 IRTree 索引过大、可扩展性低,难以处理较大规模数据(从实验中也看出了这点).

针对 IRTree 的局限性,本文提出一种新的混合索引结构 GRTree.该方法的基本思想是在语义层面针对空间语义图进行层次化划分,并将每个划分对应的子图维护在一棵树上.针对任一树节点,GRTree 一方面维护关键词以及顶点到对应子图中空间顶点语义距离的倒排索引,方便估计查询和对应子图在语义距离上的界;另一方面,通过维护节点包含空间顶点的区域,可以估计空间距离的界.这部分索引主要用于在搜索过程中较早过滤无关结果,所以我们称为过滤索引.需要注意的是,在过滤索引中维护的关键词倒排仅包含少量的关键词以及部分顶点,从而可以解决 IRTree 索引过大的问题,具体参见 2.3.2 节.

当访问到树的叶节点时,我们需要对该叶节点中包含的所有空间顶点逐个验证其排序分数.其中空间距离可直接计算,但语义距离的直接计算需要在整个图中运行最短路算法,无法保证实时性.为此我们引入了验证索引,具体介绍参见 2.3.1 节.

2.2 GRTree 结构

首先引入以下定义:

定义 1. 图的划分.给定图 $G^*=(V_P^* \cup V_N^*, E^*)$,集合 $P(G^*)=\{G_1=(V_{1P} \cup V_{1N}, E_1), \cdots, G_m=(V_{mP} \cup V_{mN}, E_m)\}$ 构成 G^* 的一个划分,如果:

1) $\forall G_k \in P(G^*), V_{kP} \subseteq V_P^*, V_{kN} \subseteq V_N^*, E_k \subseteq E^*$;

2) $\forall i \neq j, V_{iP} \cap V_{jP} = \varnothing, V_{iN} \cap V_{jN} = \varnothing$ 且 $\bigcup_{i=1}^{m} V_{iP} = V_P^*$, $\bigcup_{i=1}^{m} V_{iN} = V_N^*$;

3) $v_i, v_j \in (V_{kP} \cup V_{kN})$, $(v_i, v_j) \in E_k$ 当且仅当 $(v_i, v_j) \in E^*$.

定义 2. 划分子图.子图 $G_i \subseteq G^*$ 称为 G^* 的划分子图,如果存在 G^* 的一个划分 $P(G^*)$,使得 $G_i \in P(G^*)$.

例如在图 2 中,G_1 和 G_2 构成 G 的一个划分,二者都是 G 的划分子图.

定义 3. GRTree 是一棵平衡搜索树,满足以下性质:

1) 每个节点对应空间语义图的一个划分子图;

2) 每个节点所有子节点中的子图构成该节点中子图的划分,且不同子节点中顶点数量以及空间顶点的数量相差最多为 1;

3) 根节点对应整个空间语义图;

4) 每个非叶节点包含 m 个子节点;

5) 每个叶节点的子图中最多包含 φ 个顶点,不同叶节点所在层数最多相差 1.

以上性质保证 GRTree 是一棵平衡搜索树.为了便于表述,在本文中"顶点"一律表示空间语义图的图顶点,"节点"全部表示 GRTree 中的树节点.

图 2 上半部分给出了一个 GRTree 的样例,并给出了各节点与划分子图的对应关系.例如根节点 N_0 对应空间语义图 G,N_1 对应的划分子图为 G_1.

2.3 索引结构

2.3.1 验证索引

在叶节点中需要计算空间顶点与查询的语义距离从而用于计算排序分数验证,因此这部分索引仅在访问叶节点时用到.具体而言,给定叶节点 N,$G^*=(V_P^* \cup V_N^*, E^*)$ 是其中的划分子图,对 $\forall v \in V_P^*$,我们期望得到 $Sem_G(q,v)$ 的值.

首先引入一个新的概念:边界顶点.

定义 4. 边界顶点.给定 G^* 是 G 的一个划分子图,顶点 $u \in (V_P^* \cup V_N^*)$ 称为 G^* 的边界,如果 $\exists v \notin (V_P^* \cup V_N^*)$,使得 $(u,v) \in E$.令 $B(G^*)$ 表示 G^* 的所有边界的集合.

例如在图 2 所给的样例中,子图 G_2 的边界顶点为 $\{10\}$,子图 G_5 的边界顶点为 $\{6,10\}$,子图 G_6 的边界顶点为 $\{11,14\}$.

根据边界顶点的定义,如果顶点 $u \in (V_P^* \cup V_N^*)$,$v \notin (V_P^* \cup V_N^*)$,$u$ 和 v 之间的最短路必然包含至少一个 G^* 的边界顶点,那么如果关键词的顶点到空间顶点在整个图上的最短路径上包含非子图内的顶点,那么该路径至少经过一个边界顶点.具体地,给定 G^* 是 G 的一个划分子图,$v \in (V_P^* \cup V_N^*)$,定义

$$Sem_G^b(q,v) = \min_{u \in B(G^*)} (Sem_G(q,u) + dis_{G^*}(u,v)), \tag{2}$$

那么显然有

$$Sem_G(q,v) = \min(Sem_G^b(q,v), Sem_{G^*}(q,v)). \tag{3}$$

式(2)计算所有经过边界顶点的路径距离最小值,然后式(3)计算出所有路径距离最小值,即为

语义距离,从而计算排序分数.在式(2)中要用到 $Sem_G(q,u)$,$u\in B(G^*)$.我们可以预先计算所有 $Sem_G(q,u)$ 并建立索引,称为全局索引.而对于 $dis_{G^*}(u,v^*)$ 和 $Sem_{G^*}(q,v)$,因为在叶节点中子图 G^* 的规模已经很小,可以在搜索过程中直接计算.因此验证索引只需要全局索引即可.

表 2 中给出了全局索引中关键词 Hospital 和 Stroke 到图中个边界顶点的索引值.

Table 2　Sample of Global Index

表 2　全局索引样例

Vertices	Hospital	Stroke
v_1	0	4.15
v_8	2.35	3.95
v_{15}	3.44	6.5
v_6	0	7.33
v_{10}	2.35	1.80
v_{11}	1.69	5.99
v_{14}	3.73	3.95

在真实空间语义图中,边界点的规模要比整个图规模小得多.在本文的实验中,边界点的规模大约只有图规模的四分之一,同时在计算和存储全局索引时还可以作进一步优化,例如利用最短路之间的共同部分[4],从而减小全局索引的规模以及创建时间.

2.3.2　过滤索引

过滤索引主要用于在搜索过程中根据空间距离和语义距离的下界来较早过滤无关节点,因此我们会在每个节点将语义距离维护在一个关键词倒排中,即关键词索引.但倒排中我们只保留子图中包含的关键词,索引值为子图中语义距离的下界,这样既减少了索引规模,同时也减少了计算代价.然而在搜索过程中过滤需要对所有关键词均可得到在整个图中的语义距离的下界,因此我们还维护了节点中边界顶点的倒排,即边界索引.

1) 关键词索引.在索引中,G^* 包含的每个关键词 q 对应的索引值为 $M_{G*}(q)=\min\limits_{v\in V_P^*} Sem_{G^*}(q,v)$.

表 3 给出了 Hospital 和 Stroke 在各个节点中对应的索引.可以看到"Stroke"只在 2 个节点的索引中出现,而如果使用 IRTree 的索引方式,树中所有节点的索引都会包含"Stroke",这也是本文方法能极大地减小索引规模的主要原因.

Table 3　Sample of Keyword Index

表 3　各节点的关键词索引样例

Vertices	Hospital	Stroke	Vertices	Hospital	Stroke
N_1	0		N_4	0	
N_2	0	3.95	N_5	0	3.95
N_3	0		N_6	0	

2) 边界索引.对 G^* 中的每个边界顶点 $u\in B(G^*)$,我们同样增加一条索引,索引值为

$$F_{G^*}(u)=\min_{v\in V_P^*} dis_{G^*}(u,v).$$

表 4 给出了各个节点对应的边界索引,每个节点边界索引的大小即为该节点中边界顶点的个数.

Table 4　Sample of Border Index

表 4　各节点的边界索引

Vertices	Border Index
N_1	$(v_1,0)(v_{15},3.44)(v_8,1.80)$
N_2	$(v_{10},2.15)$
N_3	$(v_8,1.80)$
N_4	$(v_1,0)(v_{15},3.44)$
N_5	$(v_{10},2.15)(v_6,0)$
N_6	$(v_{11},1.69)(v_{14},3.73)$

以上 2 种索引构成过滤索引,之后我们估计该节点中空间顶点与关键词的语义距离下界为

$$\mathrm{Inf}_{G^*}(q)=\min(M^b_{G^*}(q),M_{G^*}(q)),\quad(4)$$

其中 $M^b_{G^*}(q)=\min\limits_{u\in B(G^*)}(F_{G^*}(u)+Sem_G(q,u))$.

下面我们给出证明,式(4)中的 $\mathrm{Inf}_{G^*}(q)$ 可用作关键词与节点中空间顶点的语义距离的下界.

引理 1. $\forall v\in V_P^*$,$\mathrm{Inf}_{G^*}(q)\leqslant Sem_G(q,v)$.

证明.根据式(3)和(4),如果 $Sem_{G^*}(q,v)\leqslant Sem^b_G(q,v)$,有

$$Sem_G(q,v)=Sem_{G^*}(q,v)\geqslant M_{G^*}(q)\geqslant \mathrm{Inf}_{G^*}(q),$$

否则,$Sem_G(q,v)=Sem^b_G(q,v)\geqslant M^b_{G^*}(q)\geqslant \mathrm{Inf}_{G^*}(q)$.

证毕.

在计算索引值时,$Sem_G(q,u)$ 通过 2.3.1 节提到的全局索引可以直接获得,F_{G^*} 通过边界索引获得,$M_{G^*}(q)$ 则通过关键词索引获得.

关键词索引和边界索引,如果在每个节点的子图中分别计算,会产生大量的重复计算,影响索引的创建效率.为了进一步提升索引的创建效率,我们可以只在各个叶节点的子图中直接计算索引,而对于其他节点的索引则充分利用其子节点中已建立好的索引,从而提升创建索引的效率.

3 搜索算法

本节主要讨论如何在 GRTree 上利用索引来搜索.算法 1 采用最佳优先策略,主要思想是根据节点的索引得出查询到子节点中所有空间顶点距离的界,从而可以在适当的情况下将所有子节点全部剪枝掉,从而提高查询速度.算法具体如下:

算法 1. 搜索算法.

输入:GRTree 的根 $root$、查询 $Q=\{\lambda,\rho,k\}$;

输出:k 个距离最近的顶点.

① 初始化优先队列;
② $Qu.Enqueue(root,0)$; /* 根节点入列 */
③ while not $Qu.isempty()$ do
④ $(N,value)\Leftarrow Qu.Deqeue()$; /* 出列 */
⑤ if N 是非叶节点 then
⑥ for each N 的子节点 N_k^c do
⑦ $Qu.Enqueue(N_k^c, \sum_{q\in Q.\lambda}\mathrm{Inf}_{G_k^c}(q))$;
⑧ end for
⑨ else if N 是叶节点 then
⑩ G^* 为 N 中的子图;
⑪ for each $v\in V_P$ do
⑫ $Qu.Enqueue(v, \sum_{q\in Q.\lambda}D(q,v))$;
⑬ end for
⑭ else
⑮ N 为下一条结果;
⑯ if 已经找到 k 条结果 then
⑰ return;
⑱ end if
⑲ end if
⑳ end while

下面我们用具体例子来描述搜索过程.由于空间距离的计算很简单,因此在此例子中我们只用语义距离代表排序分数.实际计算时,只需将语义距离替换为真实排序分数即可.

假设查询 Q,$Q.\lambda=\{\mathrm{Stroke},\mathrm{Hospital}\}$,$Q.k=1$,整个算法中优先队列的变化如下:

1) N_0 出,N_1 和 N_2 入,队列 $\{N_2,3.95\}\{N_1,4.15\}$;

2) N_2 出,N_5 和 N_6 入,队列 $\{N_5,3.95\}\{N_1,4.15\}\{N_6,7.68\}$;

3) N_5 出,顶点 v_2 和 v_6 入,队列 $\{N_1,4.15\}\{v_2,7.33\}\{v_6,7.33\}\{N_6,7.68\}$;

4) N_1 出,N_3 和 N_4 入,队列 $\{N_4,4.15\}\{N_3,5.75\}\{v_2,7.33\}\{v_6,7.33\}\{N_6,7.68\}$;

5) N_4 出,顶点 v_1 和 v_4 入,队列 $\{v_1,4.15\}\{N_3,5.75\}\{v_2,7.33\}\{v_6,7.33\}\{v_4,7.34\}\{N_6,7.68\}$;

6) v_1 出,v_1 是空间顶点,v_1 即为 top-1 结果.

4 实验结果与分析

4.1 实验设置

1) POI 数据集.实验使用 USA[5] 空间数据集.该数据集包含 100 万个美国的 POI 点,每个 POI 点由经纬度和文本描述 2 部分组成.其中各个 POI 点的文本描述不仅包含 POI 点的名称,还包含从万维网知识库中抽取的详细文本说明.

2) 空间语义图.实验利用 Freebase① 作为对 POI 数据集进行语义扩充的知识库.Freebase 是目前覆盖真实实体规模最大的知识库之一,总共包含 22 985 650 个实体.本文利用 1.1 节介绍的方法将 POI 与 Freebase 中的实体进行匹配,并生成空间语义图,最终共有 82 万个 POI 点匹配到 Freebase 中的实体.生成的图包含 87 万个顶点和 128 万条边,顶点中包含 82 万个空间顶点、22 万个边界顶点.在本文实验中,顶点之间边的权重定义为 $len(v_i,v_j)=\ln(D(v_i)D(v_j))$,其中 $D(v_i)$ 表示 v_i 的度.

3) 搜索质量评价指标.搜索质量评测旨在衡量 S^3 搜索到的结果与用户查询意图的相关程度.这个相关程度越高,说明结果越能够令用户满意.为了度量相关性,我们采用了信息检索领域广泛使用的评价指标 nDCG(normalized discounted cumulative gain).具体操作的方法如下:实验选取一组空间关键词查询(详见 4.2 节)作为测试集,对测试集中的每一条查询,通过人工标注确定相关的结果,并对每一条结果标注相关级别(级别越高表示结果越相关);之后,针对 S^3(或其他对比方法)返回的 k 条结果,根据这些结果的相关级别和排序位置聚合出 nDCG 指标(我们采用标准的 nDCG 计算方法,详见文献[6]);然后以测试集中所有查询的 nDCG 平均值作为衡量指标.直观上讲,nDCG 一方面反映了搜

① http://www.freebase.com

索的召回情况——召回的相关结果越多,nDCG 指标越高;另一方面也反映了结果的排序——相关级别高的结果越排在前面,nDCG 指标越高.

4) 搜索性能评测指标. 搜索性能评测一方面考察 S^3 构建 GRTree 索引的时间和空间开销;另一方面测试 S^3 是否能够即时返回搜索结果. 此外,实验还分析了参数对搜索性能的影响. 注:实验中涉及的所有索引均是基于磁盘(disk-based)的索引.

5) 实验环境. 本文所有程序使用 C++实现,环境为 SUSE Linux,Intel® Xeon® E5-2679, 260 GB 内存.

4.2　搜索质量评测

实验选取了 10 个有代表性的查询意图,并根据查询意图人工构造查询关键词、选定空间位置. 表 5 列出了构造的 10 条查询,可以看到,其中既包含简单语义的查询如"Hospital"(查找距离较近的医院),也包含复杂语义的查询如"Stroke Public Hospital"(查找能够治疗中风的公立医院),同时还覆盖了多个领域例如医院、餐饮. 之后,人工筛选出与每条查询相关的所有结果,并根据每条结果与查询意图的相关性以及与查询位置的空间距离标注其相关级别,用于计算 nDCG.

Table 5　Query Set for Evaluation of Effectiveness

表 5　搜索质量评测查询集

Query	Query
Q1:Hospital	Q6:Public Hospital
Q2:Stroke Hospital	Q7:Barbecue
Q3:Neurology	Q8:Chinese Food
Q4:Stroke Public Hospital	Q9:Italian Pizza
Q5:Neurology Hospital	Q10:Stroke Neurology Medical

本文使用传统空间关键词搜索方法 Traditional SKS(traditional spatial keyword search)与 S^3 进行对比. 传统的空间关键词搜索方法直接度量查询关键词与 POI 文本描述之间的 TF-IDF 相似性,并结合空间距离计算排序分数进行搜索.

需要注意的是:以上 2 种方法都需要设定查询中空间距离的比重(即式(1)中的 α). 在实验中,我们选取 $\alpha=0.8$,此时搜索方法能够取得最好的 nDCG 值. 下文也将以 0.8 设定为 α 的默认值.

图 3 给出了 2 种方法在不同结果数(即不同的 k 值)下的搜索质量. 可以看出,本文提出的 S^3 显著地优于传统空间关键词搜索方法 Traditional SKS——其 nDCG 值几乎是后者的 2 倍. 此外,随着结果数量的增加,S^3 的 nDCG 逐渐下降,这反映了在 S^3 返回的结果中排序越靠前的相关级别越高.

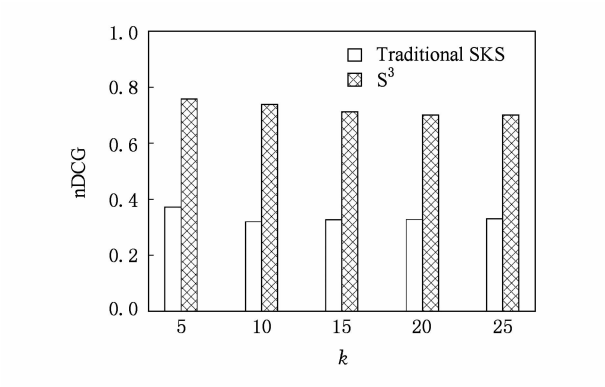


Fig. 3　Comparison of effectiveness.

图 3　搜索质量对比

S^3 显著提高搜索质量的原因主要有 2 点:1)语义扩充能够有效捕捉复杂的查询语义. Traditional SKS 方法尽管对 POI 的文本进行了扩充,但它难以为涉及多个实体的复杂查询返回正确结果. 例如表 5 中的查询 Q4,因为很多有神经科但文本中未出现"Stroke"的医院也符合要求. 然而,借助于空间语义图,S^3 可以有效地分析出复杂查询关键词之间的语义关联,从而返回更为相关的结果. 2) S^3 通过排序模型避免返回语义无关的结果. 例如表 5 中的查询 Q10,在仅考虑文本相似性时,许多语义并不准确的结果也被排在前面.

综上,搜索质量评测的结果表明,语义增强的空间关键词搜索不仅可以比传统空间关键词搜索召回更多的相关结果,而且可以提供更优的结果排序.

4.3　搜索性能评测

IRTree 方法对与某一 POI 节点相关的所有关键词都预先计算语义距离并建立索引,之后借鉴传统空间关键词搜索的策略回答查询. 本节将对 IRTree 方法和 S^3 的搜索性能进行对比.

4.3.1　参数 φ 的影响

首先分析参数 φ,即 GRTree 叶节点对应的子图中最多包含的顶点个数,对搜索性能的影响.

表 6 给出了实验结果:一方面,搜索速度随着 φ

Table 6　The Influence of φ on Search Performance

表 6　φ 对搜索性能的影响

φ	Search Time/ms	Index Size/GB	Index Time/s
50	19.5	32.9	1452
100	23.4	26.8	1389
150	25.0	24.7	1338
200	36.2	23.1	1274

的增加而降低,如 $\varphi=200$ 与 $\varphi=50$ 相比,搜索时间几乎增加了 1 倍. 原因在于:当访问 GRTree 叶节点时,S^3 依次扫描相应子图的所有顶点,分别计算排序分值并排序. 因此,叶节点子图顶点数越多,需要计算的顶点数也越多,从而导致搜索速度变慢.

另一方面,随着 φ 的增加,无论是索引规模还是索引时间,都在逐渐减小. 原因在于:当 φ 增加时,GRTree 中树节点的数量便会减少. 考虑到 S^3 对每个树节点都会建立一个索引,树节点的减少便会导致索引规模的减小,而建索引的时间也会随之减少.

参数 φ 的增加尽管可以减小索引代价,但同时会影响搜索速度. 综合考虑,本文以 $\varphi=100$ 作为默认值.

4.3.2 索引开销评测

下面考察索引规模和索引时间 2 个方面. 图 4~5 给出了空间语义图在不同规模时的实验结果,其中横坐标为顶点个数.

1) 比较索引规模. 从图 4 可以看出,S^3 的索引规模显著优于 IRTree,而且随着图规模的增加 S^3 索引规模增加的幅度远小于 IRTree. 特别是,当图的顶点接近百万时,S^3 索引规模要优于 IRTree 一个数量级. 因此,相对于 IRTree,S^3 在索引规模有着显著的优势并且具有良好的可扩展性,主要原因在于 GRTree 中每个节点的索引只保存该节点对应的子图中出现的关键词,数量要比 IRTree 少得多,从而减小了整个索引的规模.

2) 比较索引时间. 通过图 5 可以看出,与 IRTree(索引时间最大为 31 970 s)相比,S^3 构建索引的时间(最大为 1 000 s 左右)减少了一个数量级,体现了 S^3 索引的高效性.

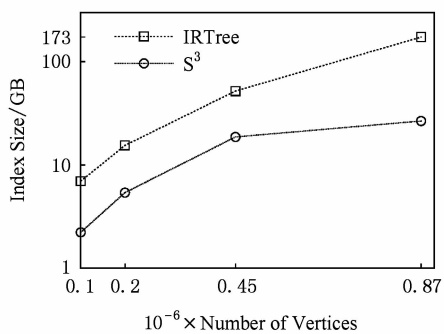


Fig. 4 Index size.

图 4 索引规模的比较

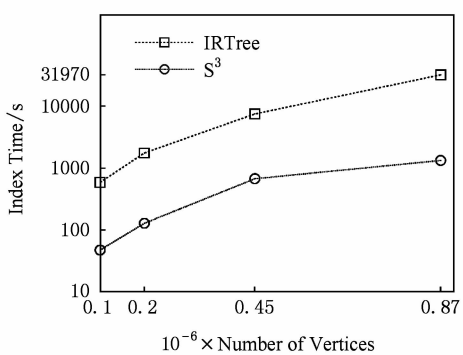


Fig. 5 Index time.

图 5 索引时间的比较

4.3.3 搜索效率评测

最后评测 S^3 的搜索速度:在 87 万顶点的语义图中对比 S^3 与 IRTree 的搜索效率和搜索可扩展性. 搜索速度通过 10 000 条查询的平均时间衡量. 默认情况下这些查询同时包含单关键词和多关键词查询.

图 6 给出了 top-k 结果数量不同时搜索效率的比较. 可以看出,在 k 取不同值时,S^3 的搜索时间显著小于 IRTree——仅为后者的一半左右. 此外,S^3 能够在 30 ms 以内完成查询,可以很好地满足即时性的搜索需求. 这主要是由于 GRTree 中将语义相近的实体放在同一个节点,所以能较早剪掉语义相关性低的结果. 在以下的实验中,我们将取 $k=10$ 为默认值.

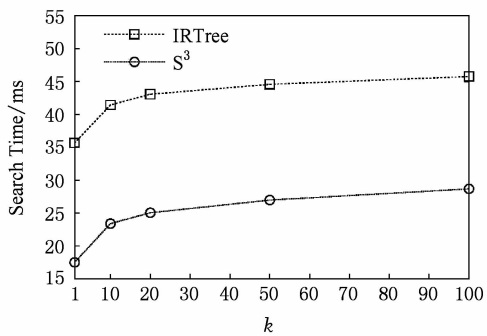


Fig. 6 Search time of varying k.

图 6 不同 k 的搜索速度

图 7 给出了当语义距离所占的权重 α 不同时搜索效率的比较. 可以观察到一个有趣的结果:随着 α 的增加,S^3 搜索时间相应地减少;而与此相反的是,IRTree 的搜索时间却在相应地增加. 这种差异主要源于 2 种方法不同的剪枝策略. GRTree 侧重于语义距离的剪枝——通过对空间语义图的层次化划分,有效地估计语义距离的界,并将语义距离过远的子图尽早剪掉. 这种策略决定了当 α 越大时,剪枝效果越明显. 与此相反,IRTree 更侧重于空间距离的剪枝——通过对空间进行层次化划分,尽早地剪掉空间距离较远的部分. 因此,当 α 越大,空间距离的

比重越小,剪枝效果越不明显.另一方面,从2种方法的比较可以看到:当α很小时,2种方法的搜索速度相近;而随着α的增加,S^3逐渐体现出其优越性;在$\alpha=0.8$时,S^3的搜索速度相比IRTree提高了3倍.

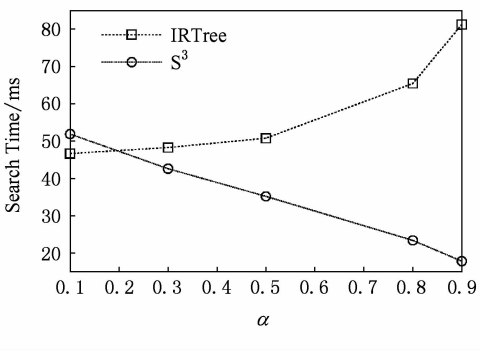


Fig. 7　Search time of varying α.

图7　不同α的搜索速度

图8给出了查询中关键词个数不同时的搜索效率比较.可以看到,搜索效率随着关键词个数增加而提升.因为在我们的模型中关键词越多,相关候选结果便越少,从而减少计算距离的代价、提升搜索速度.

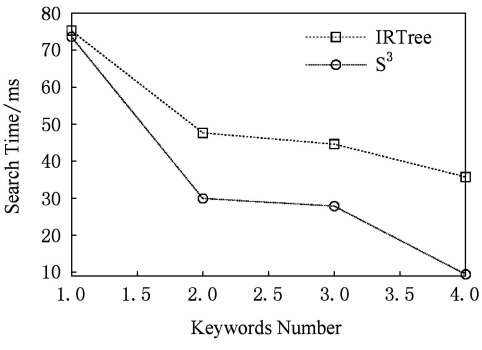


Fig. 8　Search time of varying queries.

图8　不同查询的搜索速度

图9评测了搜索效率的可扩展性:在空间语义图的顶点数量变化时,相比于IRTree,S^3在搜索效率上的优势十分显著,而且随着图规模的增加,S^3的搜索时间增加的趋势要明显缓于IRTree,这说明S^3有着更好的可扩展性.

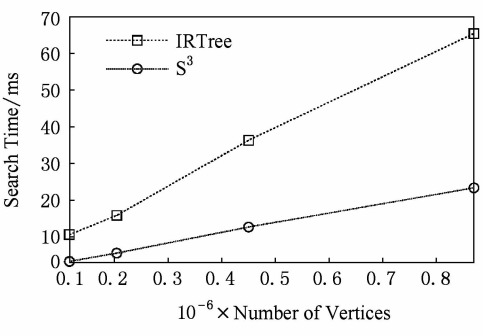


Fig. 9　Search time of varying graphs.

图9　不同图的搜索速度

5　相关工作

当前语义度量最常用的方法均利用知识库构造的图来完成[1-2,7].常用的方法包括最低公共祖先[8-9]、最短路长度[10]等.现有的基于图的关键词搜索方法为通过关键词筛选子图,然后以路径长度[11]或者子图直径[12-13]作为标准对候选子图进行排序,然而这些方法均未考虑空间距离,也很难和空间索引融合.另外有不少工作研究在路网构造的图上进行搜索,然而这些工作只考虑k近邻查询[14-15],即使加入关键词,也只考虑关键词的文本相似性[16],该相似性与图无关,而在我们的工作中相似性需要通过图上的距离来度量.

现有的空间关键词搜索工作均为同时考虑文本的相似性以及空间的距离进行搜索,需要同时考虑文本和空间2部分索引.早期索引方式分别对空间和文本单独建索引[17],为了提高搜索效率,近期工作多考虑混合索引[18].文献[3]提出IRTree结构,在IRTree的每个节点上增加文本倒排,文献[19-20]则是首先建立文本倒排,然后对每个关键词的列表建立空间索引.如果用传统文本倒排的方式建立图索引存在索引太大的问题,本文提出的索引结构GRTree则很好地解决了可扩展性问题,并且使搜索效率更高.

6　总　　结

本文提出语义增强的空间关键词搜索问题,相比传统的基于文本的方法更能满足用户的查询意图.同时本文提出一种新的索引结构GRTree,可极大地减少索引规模,提升搜索效率.

参　考　文　献

[1] Budanitsky A, Hirst G. Evaluating wordnet-based measures of lexical semantic relatedness [J]. Computational Linguistics, 2006, 32(1):13-47

[2] Maguitman A G, Menczer F, Roinestad H, et al. Algorithmic detection of semantic similarity [C] //Proc of the 14th Int Conf on World Wide Web (WWW2005). New York: ACM, 2005: 107-116

[3] Cong G, Jensen C S, Wu Dingming. Efficient retrieval of the top-k most relevant spatial Web objects [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 337-348

[4] Xiao Yanghua, Wu Wentao, Pei Jian, et al. Efficiently indexing shortest paths by exploiting symmetry in graphs [C] //Proc of the 12th Int Conf on Extending Database Technology (EDBT2009). New York: ACM, 2009: 493-504

[5] Fan Ju, Li Guoliang, Zhou Lizhu, et al. Seal: Spatio-textual similarity search [J]. Proceedings of the VLDB Endowment, 2012, 5(9): 824-835

[6] Wang Yining, Wang Liwei, Li Yuanzhi, et al. A theoretical analysis of nDCG type ranking measures [C] //Proc of the 26th Annual Conf on Learning Theory (COLT2013). New York: ACM, 2013: 25-54

[7] Huang Rui, Shi Zhongzhi. A new approach to heterogeneous semantic search on the Web [J]. Journal of Computer Research and Development, 2008, 45(8): 1338-1345 (in Chinese)
(黄瑞,史忠植. 一种新的 Web 异构语义信息搜索方法 [J]. 计算机研究与发展, 2008, 45(8): 1338-1345)

[8] Resnik P. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language [J]. Journal of Artificial Intelligence Research, 1999, 11: 95-130

[9] Couto F M, Silva M J. Disjunctive shared information between ontology concepts: Application to gene ontology [J]. Journal of Biomedical Semantics, 2011, 2(1): 1-16

[10] Ferlez J, Gams M. Shortest-path semantic distance measure in wordnet v2.0 [J]. Information Society, 2004, 28(4): 381-386

[11] Qiao Miao, Qin Lu, Cheng Hong, et al. Top-k nearest keyword search on large graphs [J]. Proceedings of the VLDB Endowment, 2013, 6(10): 901-912

[12] Pan Yifan, Wu Yuqing. Rou: Advanced keyword search on graph [C] //Proc of the 22nd ACM Int Conf on Information and Knowledge Management (CIKM2013). New York: ACM, 2013: 1625-1630

[13] Li Guoliang, Ooi B C, Feng Jianhua, et al. EASE: An effective 3-in-1 keyword search method for unstructured, semi-structured and structured data [C] //Proc of the 2008 ACM SIGMOD Int Conf on Management of Data (SIGMOD2008). New York: ACM, 2008: 903-914

[14] Samet H, Sankaranarayanan J, Alborzi H. Scalable network distance browsing in spatial databases [C] //Proc of the 2008 ACM SIGMOD Int Conf on Management of Data (SIGMOD2008). New York: ACM, 2008: 43-54

[15] Zhong Ruicheng, Li Guoliang, Tan K L, et al. G-tree: An efficient index for KNN search on road networks [C] //Proc of the 22nd ACM Int Conf on Information and Knowledge Management (CIKM2013). New York: ACM, 2013: 39-48

[16] Rocha-Junior J, Nørvåg K. Top-k spatial keyword queries on road networks [C] //Proc of the 15th Int Conf on Extending Database Technology (EDBT2012). New York: ACM, 2012: 168-179

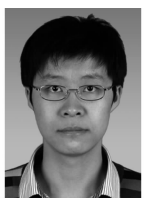
[17] Chen Y Y, Suel T, Markowetz A. Efficient query processing in geographic Web search engines [C] //Proc of the 2006 ACM SIGMOD Int Conf on Management of Data (SIGMOD2006). New York: ACM, 2006: 277-288

[18] Felipe ID, Hristidis V, Rishe N. Keyword search on spatial databases [C] //Proc of the 24th Int Conf on Data Engineering (ICDE 2008). Piscataway, NJ: IEEE, 2008: 656-665

[19] Rocha-Junior J, Gkorgkas O, Jonassen S, et al. Efficient processing of top-k spatial keyword queries [G] //LNCS 6849: Proc of the 12th Int Symp on Spatial and Temporal Databases (SSTD2011). Berlin: Springer, 2011: 205-222

[20] Zhang Dongxiang, Tan K L, Tung A. Scalable top-k spatial keyword search [C] //Proc of the 16th Int Conf on Extending Database Technology (EDBT2013). New York: ACM, 2013: 359-370

Han Jun, born in 1986. Received his PhD degree from Tsinghua University in 2015. PhD candidate in Tsinghua University. His main research interests include Web information extraction, spatial keywords search and query processing.



Fan Ju, born in 1984. Received his PhD degree from Tsinghua University in 2012. Research fellow in the National University of Singapore. His main research interests include crowdsroucing-powered data analytics, spatial-textual data processing, database usability.

Zhou Lizhu, born in 1947. Received his MSc degree from the University of Toronto of Canada in 1983. Professor and PhD supervisor in Tsinghua University. Senior member of China Computer Federation. His main research interests include database systems, query processing, and Web information extraction and mining.