北京大学学报(自然科学版)
Acta Scientiarum Naturalium Universitatis Pekinensis
doi: 10.13209/j.0479-8023.2016.008

基于空间短文本对象的检索策略

顾彦慧[1,2,†] 王道胜[3] 王永根[1,2] 龙云飞[4] 蒋锁良[1,2] 周俊生[1,2] 曲维光[1,2]

1. 南京师范大学计算机科学与技术学院，南京 210023; 2. 江苏省信息安全保密技术工程研究中心，南京 210023; 3. 南京师范大学文学院，南京 210097; 4. 香港理工大学电子计算学系，香港; † E-mail: gu@njnu.edu.cn

摘要 针对传统空间文本检索策略中的性能问题，对如何从给定的空间文本对象集合中快速有效地检索出top-k 个近似结果进行研究。基于一个空间检索的通用框架，提出一种基于空间文本对象的快速策略，用于满足用户对效率与有效性的要求，实验结果证明该策略优于现有方法。

关键词 空间文本对象; 语义相似; top-k

中图分类号 TP391

Similar Spatial Textual Objects Retrieval Strategy

GU Yanhui[1,2,†], WANG Daosheng[3], WANG Yonggen[1,2], LONG Yunfei[4], JIANG Suoliang[1,2], ZHOU Junsheng[1,2], QU Weiguang[1,2]

1. School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023; 2. Jiangsu Research Center of Information Security & Privacy Technology, Nanjing 210023; 3. School of Chinese Language and Culture, Nanjing 210097; 4. Department of Computing, The Hong Kong Polytechnic University, Hong Kong; † E-mail: gu@njnu.edu.cn

Abstract Based on the performance issue of traditional simiar spatial textual objects retrieval, a semantic aware strategy which can effectively and efficiently retrieve the top-k similar spatial textal objects has been proposed. The efficient retrieval strategy which is based on spatial textual objects has been built validated on a common framework of spatial object retrieval, and it can satisfy the efficiency and effectiveness issues of users. Extensive experimental evaluation demonstrates that the performance of our proposal outperforms the state-of-the-art approach.

Key words spatial textual object; semantic similar; top-k

在基于位置的应用中，检索相似的空间文本对象是一个重要的研究课题，例如：街旁、Foursquare 服务等，这里与位置信息有关的内容得到利用。在某个地图上，对于一个查询 Q，检索系统需要找到与之最相关的空间文本对象集合，对于集合中的每一个元素，要同时考虑空间最相近并且语义最相似。在最近的研究中，有很多研究致力于检索空间文本对象。总的说来，空间对象的索引方式可以归纳为以下几类：1) 基于 R-tree 的策略[1-4]；2) 基于网格的策略[5]；3) 基于空间填充曲线的策略[6−7]。对于文本信息，主要使用基于倒排文件的策略[8-9]和基于签名文件的策略[10]。为了检索相似的空间文本对象，一种直接的方法就是融合空间索引以及文本索引。根据融合的策略，可以大体分为面向空间索引的策略[2]和面向文本索引的策略[11]。然而，这些策略是一种松散的策略，只是把空间索引和文本索引简单地融合在一起，空间消耗比较大，检索的效率也不高。为了克服松散融合的不足，文献[7–9]提出紧密融合的策略。

由于紧密融合策略无缝地将两种不同的索引(空间索引和文本索引) 组合到一个统一的框架中，因此在这个框架中，每个顶点是两种相似度的组合

国家自然科学基金(61272221，61472191)、国家社会科学基金(11CYY030，10CYY021)、江苏省社会科学基金(12YYA002)和江苏省高校自然科学基金(14KJB520022)资助

收稿日期: 2015−06−19; 修回日期: 2015−09−03; 网络出版时间: 2015-09-30 16:02:30

(空间相似度与文本相似度)。文献[7-9]中引入了一个新型的索引结构, 称为 IR-tree, 每个 R-tree 的顶点 n 与 n 的子树对象内容的概要相关联。然而, 对于空间文本对象来说, 这种文本描述往往很短, 并且有时候几乎没有相同词存在。对于空间文本对象来说, 文本信息比较短, 假如使用传统的词频计算的方式来计算文本之间的相似度, 不能得到精确的结果[12-13], 因此传统的基于词频计算策略[7,9]不太适合用于计算文本之间的相似度。在本文中, 考虑到空间文本对象的实际情况, 提出一种基于语义相似的文本相似度计算策略来融合相应的空间信息, 从而得到比较精确的检索结果。

本文提出一种快速有效的检索相似空间文本对象的策略, 其中心思想在于建立一个完整的结构, 能无缝地融合空间索引和文本索引。通过实验可以看到, 由于考虑了空间文本对象文本的属性, 本文提出的策略能有效地提高空间文本对象的检索精度, 并且能保持较高的速度。

1 问题的提出

1.1 准备知识

假设 O 为一空间文本数据集合。每个空间对象$o \in O$被定义为一个对象对$(o.\rho, o.\varphi)$, 其中 $o.\rho$ 是一个二维的地理位置(可以用经度和维度表示), $o.\varphi$ 是一个文本描述, 也就是用户在某个地理位置的文本表达。算法要解决的问题是, 检索出与查询最相似的 top-k 个空间文本对象。对于一个查询 $Q = \langle Q.\rho, Q.\varphi, k\rangle$, 在给定的空间文本对象集合 O 中寻找一组 k 个对象 P, 这些对象按照相似度值进行排序(这里相似度值同时考虑了空间相似度以及文本相似度), 也就是说, $\forall o \in P$ 以及 $\forall o \in (O-P)$, 满足 $\mathrm{Sim}_{\mathrm{dist}}(Q,o) \leqslant \mathrm{Sim}_{\mathrm{dist}}(Q,r)$。具体地, 对于某查询, 对象 P 的排序值可以用下式计算:

$$\mathrm{Sim}_{\mathrm{dist}}(o,Q) = \alpha \cdot S_S(o.\rho, Q.\rho) + (1-\alpha)\cdot S_T(o.\varphi, Q.\varphi),$$

这里 $S_S(o.\rho, Q.\rho)$是对象 $o.\rho$ 与 $Q.\rho$ 之间的欧氏距离, $S_T(o.\varphi, Q.\varphi)$是这两个对象之间的文本距离, 这里 $\alpha \in [0,1]$。

1.2 空间临近与文本相似

空间临近 S_S 被定义为标准化的欧氏距离, 即

$$S_S = \frac{\mathrm{dist}(o.\rho, Q.\rho)}{\mathrm{dist}_{\max}},$$

这里 $\mathrm{dist}(o.\rho, Q.\rho)$是查询对象与所有数据集中对象的欧氏距离。文本之间的相似度计算可以使用现有方法, 如语言模型[7], 余弦策略[11]或 BM25[7]。因此, 文本之间的距离可以定义为

$$S_T = 1 - \mathrm{Sim}(o.\varphi, Q.\varphi)。$$

从前面的分析可以得知, 现有的基于词频以及共现的模式需要有很多相同的词出现在相似的文本中, 这种方式不适合计算两个短文本之间的相似度, 因为两个短文本之间虽然语义相似, 但往往只有很少的共同的词[12-13]。为了克服短句的不足, 很多研究致力于解决短文本相似度的计算问题, 总的来说可以分为基于知识的策略[13]、基于语料库的策略[12]、基于语法的策略[12]以及混合策略[12,14]。

为了计算文本之间的相似度 $S_T(o.\varphi, Q.\varphi)$, 我们使用最新的短文本相似度计算方法来组合不同的相似度计算策略[12,14]。需要强调的是, 短文本由词组成, 因此计算短文本间的相似度归根到底是计算词与词之间的相似度[14-15]。

1.3 通用框架模型

有很多工作致力于研究快速检索 top-k 空间文本对象, 本文在其中选取一个最具代表性的工作[1]作为研究对象。文献[1]中使用一个混合索引结构 IR-tree, 融合了空间信息以及文本信息。IR-tree 本质上是一棵 R-tree, 每个顶点关联一个倒排文件索引, 这个索引包含此顶点所有子树的信息。IR-tree 的每个顶点能总结此顶点所有子节点的文本信息, 即此顶点能描述子节点的所有信息。同时, 此顶点能估计出查询与所有在子树中对象相关程度的一个边界。因此, top-k 检索的结果使用最佳优先搜索和优先队列来实现。

2 本文提出的策略

本文考虑到文本的语义相似性, 提出一种快速有效的空间文本对象的检索方法。在介绍本文的策略之前, 首先讨论现有的集中空间文本对象的检索方法。

2.1 基准策略

若要快速有效地得到 top-k 的空间文本对象, 主要的挑战在于如何对空间信息以及文本信息进行索引。一种简单的方法就是利用 R-tree 来计算对象的空间相近程度, 利用倒排文件的方式来计算文本的相似性[16]。然后, 把两种索引结合起来, 得到 top-k 的空间文本对象。然而, 计算数据集中所有的候选对象是非常耗时的。另一种解决方法是基于倒排文件以及 R-tree 来计算空间相近度以及文本相

似度，最终生成一个 top 候选对象的集合。然而，这种方式也比较耗时，因为在预先不知道查询的情况下，不能在第一步确定需要计算多少个候选对象来满足最终的 top-k 结果，即不能保证第一步查找的对象能满足后面的 k 值。

因此，对于检索空间文本对象，一种统一的索引结构就显得非常必要[1-2]。据我们所知，大部分的研究工作使用基于词频的方式，即根据词语出现的频度以及共现来衡量两个文本对象之间的相似程度。从前面的分析中已经得知，由于空间文本对象的文本信息比较短，上述方法对于计算两个短文本显得不合适。本文第一次将语义信息考虑到检索空间文本对象当中，以解决短文本相似度计算不准确的问题。另外，本文选取一些具有代表性的方法[1-2]作为基准策略。

2.2　基于语义的策略

本文的主要目的是将语义信息融合到索引结构中，并能很好地融合空间相近性以及语义相似性。在文献[1]中，紧密融合了倒排文件索引以及 R-tree。对于给定的两个短文本 S 和 P，Sim(S, P)与短文本中代表性的词对有关。我们设 t_1^S, t_2^S, …, t_n^S 和 t_1^P, t_2^P, …, t_m^P 是 S 和 P 的词。如果$n \le m$，这时的相似度值可以表示为

$$\mathrm{Sim}(S,P)=\sum_{i}^{n}\mathrm{Sim}_r(t_i^S,P),$$

这里 Sim_r 表示代表性词对之间的相似度，其值可以从代表性的词对之间获得。因此，为文本信息建立的倒排表必须满足以下条件：1) 所有词都必须定义；2) 与词相关的一组接续列表。然而，如果一个查询 Q 包含多个词 t_1^Q, t_2^Q, …, t_n^Q，如何从中获取相关词？由于本文提出的策略是从倒排表中获取排序列表，每个词 $t_i^Q(i\in n)$ 与排序列表有关，因此很容易应用阈值算法[17]来获得与查询 Q 最相关的文本。文献[1]中将空间索引与文本索引融合在同一索引结构中。根据索引建立的方式，本文中融合文本的语义信息。

2.2.1　索引建立

IRS 策略　IRS 策略是融合语义信息到 IR-tree 中(即使用了语义相似度的 IR-tree)。IRS 的建立是使用了在 R-tree 中普遍使用的插入操作[18]。此操作包含选择叶节点和分裂节点，并保证节点的子节点包含节点中所有的文本信息。每个节点包含一个指向倒排表的指针。在 IRS 的索引结构中，叶子节点包含一组实体，记为(O, R, $O.\varphi$)，这里 O 为数据集中的空间文本对象，R 为对象 O 的矩形边界，这里 $O.\varphi$ 是对象 O 的文本描述。在子节点中，存储指向倒排表的指针，非叶节点 R 包含一组实体，记为(D, R, $D.\varphi$)，这里 D 为子节点 R 的地址，R 为包含所有矩形的最小边界矩形，$D.\varphi$ 为此矩形的文本描述。本文融合了每个空间文本对象的语义信息，所以在倒排表中，每个词对应一组文本列表对象，这些对象按照词语文本之间相似度值的倒序排列。在 IRS 中，文本信息总结了所有子节点中的文本内容，所以能预估与查询相关的范围。

DIRS 策略　从 IRS 策略可以看出，索引建立的时候仅仅考虑了空间信息，即最小矩形是否与空间位置有关。在实际情况中，对于一个查询，离查询最近的不一定是用户最终想得到的结果，因此在索引建立的时候，需要同时考虑空间信息和文本信息。与 IRS 策略相似的是，非叶节点 R 包含一组实体，记为(D, R, $D.\varphi$)。与 R-tree 操作相同，对于一个新的空间文本对象，为了选择合适的插入路径，DIRS 同时考虑了空间信息和文本信息。在这里，记每个节点的条目为 En_1, En_2, …, En_k。令 O_{new} 是新插入的空间文本对象，在 R-tree 中，En_i 的面积由于新空间文本对象 O_{new} 的加入而变大。这里，本文使用一个衡量指标 EnlargeRec(En_k)来描述面积的增加值:

$$\mathrm{EnlargeRec}(\mathrm{En}_k)=\mathrm{Rec}(\mathrm{En}_k^{\mathrm{new}}.R)-\mathrm{Rec}(\mathrm{En}_k),$$

这里，$\mathrm{Rec}(\mathrm{En}_k^{\mathrm{new}}\cdot R)$是 En_k 融入了新空间文本对象 O_{new} 后的新实体。考虑了文本信息后，面积增加可以用下式表示:

$$\mathrm{EnlargeRec}_{\rho\varphi}(\mathrm{En}_k)=(1-\delta)\cdot\frac{\mathrm{RecEn}_k}{\mathrm{Rec}_{\max}}+\delta\cdot S_T(\mathrm{En}_k,O.\varphi),$$

这里 δ 是调节空间信息与文本信息的参数，$\mathrm{Rec}_{\max}$ 是包含所有空间文本对象的最小边界矩形。

可以看出，在 DIR 索引的建立过程中，不同于 R-tree，插入操作中，在选取子树的时候需要考虑到文本信息，也就是说，使得 $\mathrm{EnlargeRec}_{\rho\varphi}(\mathrm{En}_k)$的值最小。同样在分裂操作中，需要考虑文本的信息，使得整个索引的建立过程中同时考虑到空间信息与文本信息。

2.2.2　查询过程

最佳优先遍历算法[19]和优先队列是用于存储访问过的顶点和空间文本对象。这里用 $\min_{\rho\varphi}(Q,R)$ 表示查询 Q 与矩形 R 之间的边界，$\mathrm{Sim}_{\mathrm{dist}}(Q,o)$表示

Q 与每个空间文本对象 o 之间的距离，详细说明见表 1。需要说明的是，$\mathrm{Sim}_{\mathrm{dist}}(Q, o)$算法仅计算查询 Q 与算法遍历过的对象或矩形。这里使用一个例子来说明 DIRS 与 IRS 策略的不同，具体过程见表 2 和 3。

从表 2 可以看出，为了得到 top-2 对象 o_2, o_1, IRS 策略的检索过程遍历 R_5, R_6, R_1, R_2, R_3, R_4, 也就是遍历整棵树中大部分的节点。这是因为 IRS 策略在建立索引的时候，只考虑了空间信息，忽视了文本的语义信息。

从表 3 可以看出，DIRS 策略的检索过程仅需遍历 R_5, R_1, R_3 就能得到最终的 top 结果。这是因为 DIRS 在索引建立时考虑了文本的语义信息，使得索引结构更加准确，查询的过程只需要遍历很少的节点，即只需要 5 步，少于 IRS 策略的 7 步。

表 1　查询 Q 与对象以及矩形的距离值

Table 1　Distance value of objects and rectangles to query Q

O	$\mathrm{Sim}_{\mathrm{dist}}$ (Q, o)	R	$\min_{p\varphi}$ (Q, R)	
			IRS	DIRS
o_1	0.2614	R_1	0.2202	0.3220
o_2	0.2305	R_2	0.4204	0.2052
o_3	0.3232	R_3	0.3019	0.1716
o_4	0.5019	R_4	0.1151	0.4507
o_5	0.3953	R_5	0.0520	0.0622
o_6	0.4929	R_6	0.2749	0.5002
o_7	0.6927	R_7	0	0
o_8	0.5608			

表 2　基于 IRS 策略的查询过程

Table 2　Query processing based on IRS strategy

步骤	状态		
	入队	出队	队列
1	R_7	R_5, R_6	R_5, R_6
2	R_5	R_1, R_4	R_1, R_4, R_6
3	R_4	o_4, o_5, o_6	R_1, R_6, o_4, o_5, o_6
4	R_1	o_1, o_3	o_1, o_3, o_4, o_5, o_6, R_6
5	R_6	R_2, R_3	R_2, R_3, o_1, o_3, o_4, o_5, o_6
6	R_3	o_2	R_2, o_1, o_2, o_3, o_4, o_5, o_6
7	无	o_1, o_2	o_1, o_2
Top-k (k=2)		o_1, o_2	

表 3　基于 DIRS 策略的查询过程

Table 3　Query processing based on DIRS strategy

步骤	状态		
	入队	出队	队列
1	R_7	R_5, R_6	R_5, R_6
2	R_5	R_1, R_3	R_1, R_3, R_6
3	R_3	o_1	R_1, R_6, o_1
4	R_1	o_2, o_3	o_1, o_2, o_3, R_6
5	无	o_1, o_2	o_1, o_2
Top-k (k=2)		o_1, o_2	

3　实验分析

我们通过实验，从有效性和性能两个方面检验本文提出的策略。实验在 16-core Intel® Xeon® E5530 服务器上进行，操作系统为 Debian 2.6.26-2，所有程序使用 C 语言编写，用 GNU gcc 编译器编译。

本文使用文献[20]中使用的数据集。此数据集共有 225098 个用户，22506721 个唯一的空间文本对象(这里的对象既含有空间信息，又含有文本信息)。为了检验本文所提策略的有效性，将 IR-tree 没有融入语义信息作为基准策略，记为 baseline，将 IR-tree 使用了语义信息记为 IRS，将文本信息索引加上语义融合记为 DIRS。本文所用数据集的统计信息见表 4。

3.1　有效性实验结果

在有效性试验中，我们随机选择 5 个查询，分别计算在基准策略、IRS 以及 DIRS 三种策略下的精度。从前面的分析可以看出，影响有效性的因素有两个。一个影响因素是在计算过程中调节融合空间相近程度与文本相近程度的参数 α，这个参数的作用主要是在调节空间相近与文本相似这两个因素的权重。这是由于在实际的系统中，人们由于对空间相似度与文本相似的关心程度不一样，因此需要调节这个权重来计算出更加精确的结果。另一个影响因素是索引建立时，用来调节文本加强程度的参数 δ。由于本文所提策略是在索引建立初期就融合考虑文本因素对于整个索引建立的影响，因此 δ 就是为了调节文本因素对索引建立影响程度的参数。我们通过交叉验证实验，选测 α 和 δ 的最佳值。

α 和 δ 的取值都在 0.1~0.9 之间。从实验结果得到：对于 IRS，α=0.68；对于 DIRS，α=0.68，

表 4　数据集统计信息

Table 4　Statistics on datasets

数据集名称	数据集大小/k	数据集类型	平均文本长度/个	最小文本长度/个	最大文本长度/个
Foursquare	1	原始	7.32	5	21
		处理后	6.12	2	10
	5	原始	7.54	5	21
		处理后	6.03	2	10
	10	原始	7.29	5	21
		处理后	6.45	2	10
	20	原始	7.63	5	21
		处理后	6.10	2	10

说明: “原始”表示从原始的数据集中抽取的文本结果, 没有经过任何预处理; “处理后”表示去除停用词以及规则化后的结果。

表 5　有效性实验结果(精度)

Table 5　Experimental results on effectiveness (Precision)

策略	精度				
	查询 1	查询 2	查询 3	查询 4	查询 5
基准策略	0.24	0.22	0.28	0.32	0.18
IRS	0.78	0.82	0.86	0.62	0.76
DIRS	0.78	0.82	0.86	0.62	0.76

δ=0.2。有效性实验结果如表 5 所示, 可以看出, IRS 策略和 DIRS 策略由于考虑了语义信息, 所以明显优于基准策略。

3.2　不同参数下实验结果

由于本文的实验结果与两个重要的参数 α 和 δ 有关, 因此我们分别选取 α 和 δ 在 0.1~0.9 区间, 按照粒度 0.1 来验证, 结果如表 6 和 7 所示。可以看到, 相比于 IRS 策略, 由于 DIRS 策略在索引建立时已经考虑了文本之间的信息, 所以能大幅度减少访问时间。从表 6 和 7 还可以看出, 不同的 α 和 δ 值对效率的影响主要表现在节点访问的顺序以及在某个节点访问的时间长短上, 所以 α 和 δ 这两个参数对基准策略的影响不大, 而对 IRS 策略以及 DIRS 策略有一定的影响。因此, 为了平衡有效性与效率之间的关系, 如何选取 α 和 δ 参数值也是一个重要的研究课题。

3.3　效率实验结果

性能的比较使用有效性试验中最佳的参数配置, 通过以下的实验来检验: 不同的数据集大小以及不同的 k 值。本文从数据集中随机抽取 10 个对象作为查询, 实验结果包括平均查询时间, 结果如

表 6　不同 α 下效率实验结果

Table 6　Experimental results on efficiency under different α

α	基准策略		IRS		DIRS	
	时间/s	个数	时间/s	个数	时间/s	个数
0.1	211.5	1989	71.2	985	67.2	841
0.2	207.2	1975	72.6	991	68.8	865
0.3	212.6	1936	75.2	1002	71.5	890
0.4	203.2	1957	80.1	1021	72.3	907
0.5	201.5	1879	80.5	1212	75.6	912
0.6	212.1	1905	81.5	1202	76.5	1050
0.7	206.3	1849	82.3	1325	78.9	1101
0.8	208.2	1908	83.5	1205	80.2	1153
0.9	214.1	2001	86.5	1352	83.3	1260

表 7　不同 δ 下效率实验结果

Table 7　Experimental results on efficiency under different δ

δ	基准策略		IRS		DIRS	
	时间/s	个数	时间/s	个数	时间/s	个数
0.1	208.2	1975	69.1	892	63.2	812
0.2	209.3	1980	71.3	986	71.2	971
0.3	213.5	1939	75.6	1002	73.8	924
0.4	201.2	1968	79.5	1102	75.3	1043
0.5	206.3	1960	81.2	1203	76.6	1051
0.6	207.5	1970	83.7	1212	78.5	1103
0.7	210.1	1983	84.1	1301	80.4	1151
0.8	209.6	1979	85.2	1310	81.1	1204
0.9	212.1	1984	85.9	1321	82.1	1298

图 1 和 2 所示。可以看出，由于 DIRS 在索引建立时考虑了文本信息，使得查询访问较少的节点就能得到最终的结果，所以在性能上 DIRS 比 IRS 更优。

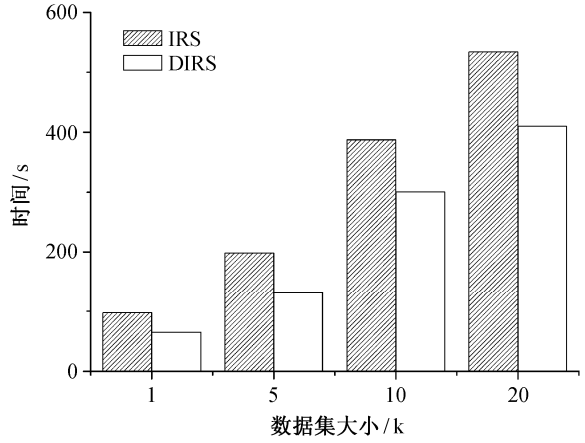

图 1　不同数据集大小结果

Fig. 1　Results of data collection size

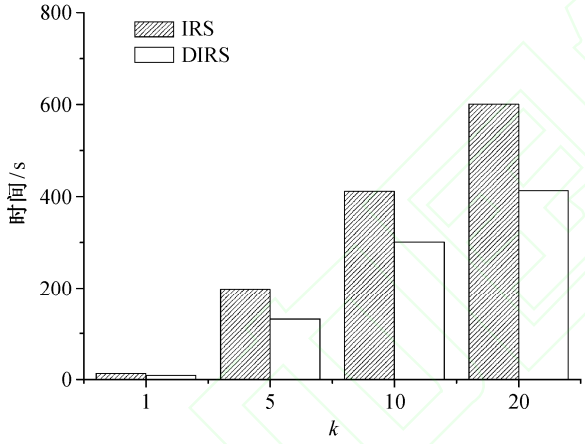

图 2　不同 *k* 值结果

Fig. 2　Results of *k* value

4　总结

本文针对传统空间文本检索策略中的有效性问题，对如何从给定的空间文本对象集合中快速有效地检索出 top-*k* 个近似结果进行研究。主要贡献如下。

1) 以空间文本对象检索为研究对象，提出一种基于语义的策略，在空间文本兑现检索过程中考虑到语义信息(这种语义信息在实际系统中非常重要)，通过建立综合的索引来无缝融合空间索引与文本索引。

2) 提出一种快速有效的空间文本对象的检索算法，这种算法对于实际应用系统来说非常重要，因为用户更加倾向于找到语义相似的对象。

3) 对实际数据的实验表明，与现有策略相比较，本文提出的策略在速度以有效性方面有较大优势。

参考文献

[1] Cong G, Jensen C S, Wu D. Efficient retrieval of the top-k most relevant spatial web objects. // Proceedings of the VLDB Endowment. Lyon, 2009: 337–348

[2] Li Z, Lee K C K, Zheng B, et al. Ir-tree: an efficient index for geographic document search. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(4): 585–599

[3] Zhang D, Chee Y M, Mondal A, et al. Keyword search in spatial databases: towards searching by document // Proceedings of IEEE International Conference on Data Engineering. Shanghai, 2009: 688–699

[4] Zhou Y, Xie X, Wang C, et al. Hybrid index structures for location-based web search // Proceedings of International Conference on Information and Knowledge Management. Bremen, 2005: 155–162

[5] Khodaei A, Shahabi C, Li C. Hybrid indexing and seamless ranking of spatial and textual features of web documents // Proceedings of International Conference on Database and Ex-pert Systems Applications. Bilbao, 2010: 450–466

[6] Chen Y Y, Suel T, Markowetz A. Efficient query processing in geographic web search en-gines // Proceedings of ACM SIGMOD International Conference on Management of Data. Chicago, 2006: 277–288

[7] Christoforaki M, He J, Dimopoulos C, et al. Text vs. space: efficient geo-search query processing // Proceedings of International Conference on Information and Knowledge Man-agement. Glasgow2011: 423–432

[8] Cong G, Jensen C S, Wu D. Efficient retrieval of the top-k most relevant spatial web objects // Proceedings of the VLDB Endowment. Lyon, 2009: 337–348

[9] Li Z, Lee K C K, Zheng B, et al. Ir-tree: an efficient index for geographic document search. IEEE Transactions on Knowledge and Data Engineering, 2011, 23(4), 585–599

[10] Felipe I D, Hristidis V, Rishe N. Keyword search on spatial databases // Proceedings of IEEE International Conference on Data Engineering. Cancun, 2008: 656–665

[11] Rocha-Junior J B, Gkorgkas O, Jonassen S, et al. Efficient processing of top-k spatial key-word queries // Proceedings of International Symposium on Spatial and Temporal Databases. Minneapolis, 2011: 205–222

[12] Islam A, Inkpen D. Semantic text similarity using corpusbased word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data, 2008, 2(2): 1-25

[13] Li Yuhua, McLean D, Bandar Z, et al. Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering, 2006, 18(8): 1138-1150

[14] Tsatsaronis G, Varlamis I, Vazirgiannis M. Text relatedness based on a word thesaurus. Journal of Artificial Intelligence Research, 2010, 37(1): 1–40

[15] Gu Yanhui, Yang Zhenglu, Xu Guandong, et al. Exploration on efficient similar sentences extraction. World Wide Web, 2014, 17(4): 595-626

[16] Martins B, Silva M J, Andrade L. Indexing and ranking in GEO-IR systems // Proceedings of the Workshop On Geographic Information Retrieval. Bremen, 2005: 31–34

[17] Fagin R, Lotem A, Naor M. Optimal aggregation algorithms for middleware // Proceedings of ACM Symposium on Principles of Database Systems. Santa Barbara, 2001: 102–113

[18] Guttman A. R-trees: a dynamic index structure for spatial searching // Proceedings of ACM SIGMOD International Conference on Management of Data. Boston, 1984: 47-57

[19] Hjaltason G R, Samet H. Distance browsing in spatial databases. ACM Transactions on Da-tabase Systems, 1999, 24(2): 265-318

[20] Cheng Z, Caverlee J, Lee K, et al. Exploring millions of footprints in location sharing ser-vices // Proceedings of International Conference on Web and Social Media. Barcelona, 2011: 81–88