

融合空间计算的 POI 地理编码方法^{*}

严 潇, 芮 维, 樊亚新, 朱欣焰, 余 冰

(武汉大学 测绘遥感信息工程国家重点实验室, 武汉 430079)

摘 要: 位置信息在日常生活和专业领域中的重要性日益凸显, 为了更加有效地处理含有空间陈述的复杂的 POI 位置描述短语, 提取其中的位置信息, 基于一种融合语义本体的对象级规则匹配方法, 提出了在传统的 POI 地理编码过程中融入空间计算, 用以支持带有空间陈述的复杂的 POI 位置查询, 并对搜索得到的结果集按照一定的标准进行排序, 并将最符合用户需求的结果可视化地反馈给用户。实验分析比较结果显示, 该方法能有效地解析匹配含有空间陈述的复杂 POI 位置描述短语, 准确快速地查询到与用户描述最为接近的 POI 数据。

关键词: 位置; POI; 地理编码; 空间计算

中图分类号: TP208

文献标志码: A

文章编号: 1001-3695(2016)09-2652-04

doi: 10.3969/j.issn.1001-3695.2016.09.020

POI encoding method fusion of spatial computation

Yan Xiao, Guo Wei, Fan Yaxin, Zhu Xinyan, She Bing

(State Key Laboratory of Information Engineering in Surveying, Mapping & Remote Sensing, Wuhan University, Wuhan 430079, China)

Abstract: The location information had become very important in daily life and areas of expertise. In order to more effectively deal with complex POI position statements contained space description phrase and extract the location information, A POI encoding method fusion of spatial computation was proposed. It was based on an object-level rule matching method fusion of semantic ontology to support c-omplex POI query. Then, we sorted the result set according to ascertain standard and feedback the optimal results to the user. The experimental results show that this method can effectively analysis and match co-mplex POI position description phrase contained spatial representations.

Key words: location; POI; geocoding; spatial computation

0 引言

近年来,随着经济社会科技的发展,位置信息的重要性在日常的位置服务(LBS)和专业领域研究中日益凸显。在基于位置的服务(LBS)的应用领域中,如旅游信息系统、交通查询系统等都需要尽可能地理理解用户输入的自然语言位置描述请求;在专业领域中,日常记录的文档材料和大量历史记录都需要进行位置的准确定位来提高办事效率,并为后续时空分析提供精度保障。

地理编码(geocoding)也称地址匹配,是指建立地理位置坐标与给定地名地址一致性的过程,以对属性数据和地理实体进行位置确定和空间检索^[1],从而实现非空间数据的上图^[2],其包括编码技术和匹配技术^[3]。其中,最主要的是匹配技术,现在最常用的匹配技术是将输入的地名地址通过中文分词、全文检索等步骤从而得到的最准确的匹配结果。中文分词是把输入的计算机不能理解的字符串或者序列按照一定的规则切分并最终组合成计算机能够理解的词序列的过程^[4]。目前国内对于中文分词主要有以下研究成果:正向最大匹配法、反向最大匹配法、分词与词性标注一体化方法、最佳匹配法、专家

系统方法、最少分词词频选择方法、神经网络方法等^[5]。全文检索技术^[6]是一种重要的关键词检索方法,是指计算机索引程序通过扫描文章中的每个词,对每一个词建立索引,当用户查询时,检索程序就根据事先建立好的索引进行查找,并将查找的结果反馈给用户的检索方式^[7]。目前国际上建立了很多全文搜索引擎,其中,Google最为著名,其是当前搜索准确度和用户查询相关度最好的全文搜索引擎。国内以百度最为著名,百度已经成为世界上最强大的中文搜索引擎。现有的中文分词与匹配技术在POI名称匹配上存在一些问题:分词词典没有考虑POI的特殊性,在匹配过程中没有区分不同切分单位在匹配过程中的重要程度,普通匹配方法没有考虑POI具有空间位置的属性^[8];现有POI搜索技术仍然是基于关键词匹配的方式,即在不同字段间查找符合输入关键字匹配要求的地理对象,无法支持复杂语义的查询^[9],虽然杜冲等人^[10-13]对地理实体及其空间关系进行了相关的研究,但也只是处于理论探索阶段,难以支撑包含空间陈述的POI查询。

针对传统POI地理编码方法中的不足,本文基于一种融合语义本体的对象级规则匹配方法,对于“武汉大学附近的医院”这类带有空间陈述的位置描述短语,将空间计算融于POI

收稿日期: 2015-04-24; **修回日期:** 2015-07-07 **基金项目:** 国家科技支撑计划位置传感网与全息位置地图关键技术及其应用(2012BAH35B03);公安部重点研究计划项目(201302ZDYJ018)

作者简介: 严潇(1989-),女,湖北咸宁人,硕士研究生,主要研究方向为自然语言解析(769964187@qq.com);芮维(1981-),男(通信作者),湖北公安人,教授,博士,主要研究方向为智慧城市时空数据流计算、自然语言位置解析、机器人导航与Kinect实时制图等;樊亚新(1990-),男,湖北天门人,在读博士,主要研究方向为警务大数据分析;朱欣焰(1963-),男,江西人,教授,博士,主要研究方向为分布式空间数据库、地球信息服务、地理信息系统与应用等;余冰(1988-),男,湖北人,博士,主要研究方向为位置自然语言理解和空间分析。

数据库检索查询过程中。具体而言,主要从语义建模、解析匹配、计算查询和排序这几个方面进行研究。a) 在分析大量的语义位置描述的基础上建立模型,并构建相应的位置本体,填充本体实例,同时根据位置概念本体建立对应的位置结构概念本体;b) 初始化语义位置模型知识库和规则匹配引擎,准备进行解析匹配:以自然语言文本位置描述语句作为输入,并利用已经构建好的语义位置模型知识库对描述语句进行位置基础概念本体实例或别名的提取工作,将原始自然语言文本位置描述语句和之前构建的位置基础概念本体实例作为输入,并利用规则匹配引擎进行匹配;c) 根据匹配得到的位置概念本体实例的显著性比较确定计算策略,进行文本空间联合索引,并对查询结果按照一定的标准进行排序显示。

1 POI 及空间关系建模

POI 即兴趣点,是 GIS 中的一个术语,泛指一切可以抽象为点的地理对象,尤其是一些与人们生活密切相关的地理实体,如学校、银行、餐馆、医院、超市等^[14]。本文将 POI 的组成成分主要划分为如图 1 所示的几个部分。

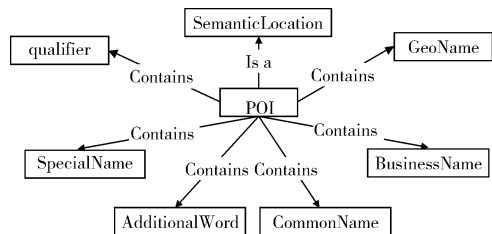


图1 POI 本体构成图

特名 (SpecialName), 即未登录词, 是指每个 POI 名称特有的, 且在数据库中不存在的; 通名 (CommonName), 即指一类地理实体的总称; 业务名 (BusinessName), 即指一类业务的总称; 修饰词 (Qualifier), 与中文里的修饰词大体一致; 地名 (GeoName), 指行政区划名称, 包括省、市、县、镇、村等; 部分 POI 名称中包含对其位置判断的补充信息, 如“华莱士广埠屯连锁店”中的“广埠屯连锁店”“连锁店”为信息辅助词。

在人们的日常生活中, 人们通常采用自然语言来表达对空间关系的理解, 称为自然语言空间关系。自然语言空间关系是空间关系在更高层次上的概念化, 它比空间关系更接近于人们的使用习惯^[15]。自然语言空间关系分类如图 2 所示。

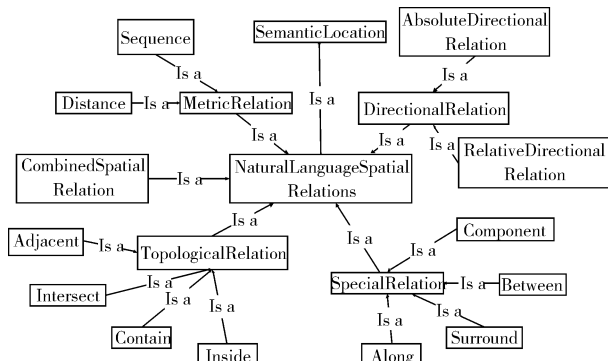


图2 自然语言空间关系分类图

本文根据人们的习惯将人们经常使用的自然语言空间关系 (NaturalLanguageSpatialRelation) 分为五类: 拓扑关系 (TopologicalRelation)、方位关系 (DirectionalRelation)、度量关系 (MetricRelation)、组合空间关系 (CombinedSpatialRelation)、

特殊空间关系 (SpecialRelation) 这几类。

2 POI 地理编码方法

针对传统地理编码的不足, 本文基于一种融合语义本体的对象级规则匹配方法, 提出将空间计算融于 POI 地理编码的过程中, 实现包含空间陈述的 POI 查询。a) 根据自然语言语义位置模型建立地名、POI 及空间关系本体, 并填充本体实例, 根据位置概念本体建立对应的位置结构概念本体, 然后初始化语义位置模型知识库和规则匹配引擎;b) 以一个 POI 位置描述短语为输入进行 POI 地理编码过程, 具体流程如图 3 所示。

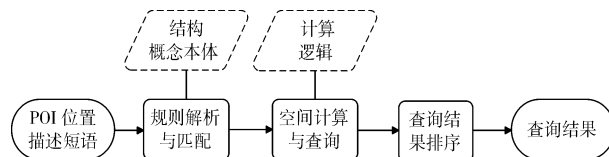


图3 POI 地理编码方法整体流程图

在初始化结束后, a) 用位置结构概念本体 (即融合语义本体的规则) 对输入的 POI 位置描述短语进行解析与匹配, 得到若干个实体概念本体对象并生成匹配树集;b) 根据对象的显著性差别制定计算策略和 POI 查询步骤;c) 进行 POI 查询过程, 并在数据库查询过程中融入空间计算;d) 对 POI 的初步查询结果进行打分排序, 将分数较高的前几个查询结果反馈给用户。

2.1 规则解析与匹配

本文中的解析匹配是基于位置结构本体 (即融合语义本体的规则) 的解析与编译, 规则的表现形式是正则表达式, 规则匹配就是正则表达式匹配。匹配抽取引擎是通过融合语义本体的规则提取和匹配出位置概念对象, 并构建出对象关系树, 即文中所说的匹配树。整体流程如图 4 所示。

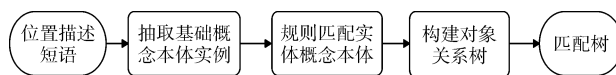


图4 规则解析匹配整体流程

据图 4 可知, a) 提取输入位置描述短语中的基础概念本体实例;b) 根据得到的基础概念本体实例和规则对描述短语进行匹配, 得到实体概念本体;c) 构建对象关系树, 输出得到匹配树。算法 1 为规则解析匹配的伪代码。

算法 1 算法 match (String text) 伪代码

输入: text; POI 位置描述短语。

输出: matchTree; 匹配树。

```
basicEntityObjs = {}, entityObjs = {}, treeList = {},
basicEntityObjs = searchWordInText (text) // 抽取基础概念本体实例
entityObjs = ruleMatch (basicEntityObjs, List<rule>) // 规则匹配
while entityObjs.spatialRelationNum = 1 do // 创建对象关系树
treeList = generateTree (basicEntityObjs, List<rule>)
matchTree = generateTree (treeList, spatialRelationObj, List<rule>)
while entityObjs.spatialRelationNum > 1 do
treeList = generateTree (basicEntityObjs, List<rule>)
for each spatialRelationObj in entityObjs do
matchTree = generateTree (treeList, matchTree, spatialRelationObj,
List<rule>)
```

以“广八路与八一路交叉口的酒店”为例, 首先对此描述语句进行基础概念本体实例的抽取, 得到“广八”“八一”“路”“交叉口”“酒店”等, 然后根据这些实例通过规则引擎对描述短语进行解析匹配, 得到匹配树, 如图 5 所示。

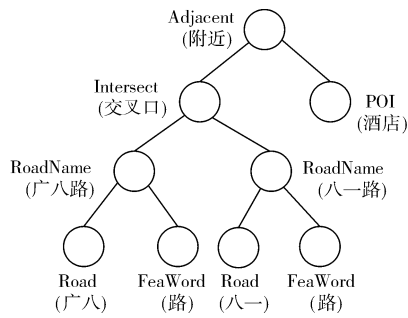


图 5 匹配树示意图

图 5 中的匹配树对应的匹配结果是:(((广八路)与(八一路)|交叉口)|附近)|酒店),其叶子节点都为基本概念本体实例,如“广八”“八一”“路”等,“酒店”同时可以被匹配成 POI;其他的节点为实体概念本体节点,具体而言,有空间关系节点、参考地物节点和目标节点。

2.2 空间计算与查询

在自然语言位置描述中,人们习惯在用 POI 描述位置的同时混合着一些自然语言空间关系的描述,从而能更加准确地表达出目标位置,如在“解放南路旁边的融科天城”“广八路与八一路交叉口附近的医院”中,“解放南路旁边”和“广八路与八一路交叉口附近”限定了“融科天城”和“医院”的位置范围。针对这些含有空间关系的 POI 位置描述短语,通过规则对其解析匹配后,本文在 POI 查询过程中对位置描述短语中的空间陈述进行了计算,将空间计算与空间数据库查询融为一体,得到了较为准确的结果集。图 6 为计算查询的整体流程。

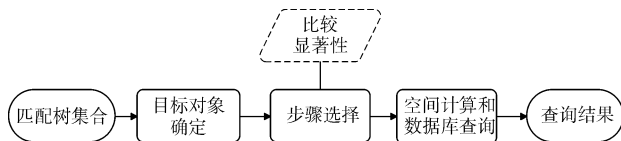


图 6 空间计算查询整体流程

如图 6 中所示,本文 a) 通过融合语义本体的规则的方法对 POI 位置描述短语进行解析匹配,得到匹配树,对匹配树进行 POI 查询计算,先确定匹配树的目标对象节点(本文是 POI);b) 比较对象的显著性大小,制定相应的计算策略,进行 POI 查询,并在查询过程中融入空间关系的计算;c) 得到查询结果集。算法 2 为空间计算查询的伪代码。

算法 2 算法 queryPOI(String text) 伪代码

输入:matchTree;规则解析匹配得到的匹配树。

输出:List<POI>;POI 对象列表。

poiObjs = {}, targetList = {}, range = {}

spatialRelationNodeNum = calculateNum(matchTree)

while spatialRelationNodeNum = 1 do //含有简单空间关系

if targetNode.significance > entityNode.significance then //显著性比较

targetList = searchPoiInDataBase(targetNode) //空间数据库查询 POI

for each target in targetList is in spatialRange(refer-ence,spatialRelationWord) do //进行空间计算

poiObjs.add(target)

else

range = calculateSpatialRelation(reference,spatialRelationWord) //进行空间计算,得到空间范围

poiObjs = searchPoiInDataBase(range) //空间数据库查询 POI

while spatialRelationNodeNum > 1 do //含有复杂空间关系

if targetNode.significance > spatialRelationNode.significance then //显著性比较

targetList = searchPoiInDataBase(target) //空间数据库查询 POI

for each target in targetList is in spatialRange(spatialRelationObj,spatialRelationWord) do //进行空间计算

poiObjs.add(target)

else //显著性比较

range = calculateSpatialRelation(spatialRelationObj,s-spatialRelationWord) //进行空间计算,得到空间范围

poiObjs = searchPoiInDataBase(range) //空间数据库查询 POI

2.3 查询结果排序

通过空间计算查询得到的结果集 poiObjs 后,需要对其按照一定的标准方法进行排序,然后将筛选排序后的结果可视化地向用户展示。在本文中,对结果集中的 POI 数据进行打分,并按分数从高到低对结果进行排序。

结果集中 POI 的分数是根据距离(Dis)与语义相似度(S)来加权得到的,公式为

$$\text{Core} = \frac{1}{\text{Dis}} \times w_1 + S \times w_2 \quad (1)$$

这里距离(Dis)是指结果集中 POI 与参考地物间的距离;语义相似度(S)是指两个 POI 的相似度,即结果集中的 POI 与输入 POI 的相似度; w_1 、 w_2 为距离与语义相似度在打分时所占的权重,代表着重要程度。

本文中,POI 由通名(CommonName)、特名(SpecialName)、业务名(BusinessName)、修饰词(Qualifier)、地名(GeoName)、信息辅助词(AdditionalWord)等构成。在计算语义相似度(S)时,就是比较结果集中的 POI 这些组成部分与输入 POI 的相似程度。具体公式为

$$S = \frac{p_c \times w_c + p_s \times w_s + p_b \times w_b + p_q \times w_q + p_g \times w_g + p_a \times w_a + \dots}{w_c + w_s + w_b + w_q + w_g + w_a + \dots} \quad (2)$$

其中: p_c 、 p_s 、 p_b 、 p_q 、 p_g 、 p_a 、... 它们的值为 0 或 1,当结果集中 POI 的某部分与输入 POI 的相应部分相同时,其对应的 p 值为 1,否则, p 值为 0; w_c 、 w_s 、 w_b 、 w_q 、 w_g 、 w_a 、... 为 POI 各组成部分的权重,代表着各组成部分的重要程度,本文中, w 值是专家根据 POI 名称各部分的重要性的和人们的习惯进行评价打分所得,并根据实际情况进行调整。

例如,“解放南路旁边的融科天城”通过一系列的计算查询后得到的结果集中有多个名称为“融科天城”的 POI 数据,其语义相似度一样,分数主要取决于结果集中的“融科天城”与解放南路之间的距离,距离越小,分数越高,排序便越靠前。

3 原型系统及实验

针对含有空间陈述的复杂 POI 位置描述的提取和定位,设计并实现了泛在位置信息提取和定位原型系统,系统前台使用 Flex 富客户端实现,后台使用 Java 实现,前后台通信利用 Servlet 接口实现。系统可支持单条语句的空间配准,也支持 Excel 文件批量上传空间配准。原型系统界面如图 7 所示。

图 7 显示原型系统的前台界面,标志 1 处为位置描述短语的输入文本框,标志 4 处为查询结果显示框,其结果按分数从高到低进行排序显示,当鼠标移动到地图上的某个查询结果点位处会显示标志 3 处的悬浮说明框,其中包含了查询结果的中文描述、此查询结果在排序上的分数以及此查询结果的概念本体组成结构。

以“广八路与八一路交叉口的酒店”为例,查询结果在地图上以半透明的红色小圆圈表示,在标志 4 处显示了一系列的查询结果,其中“丰颐大酒店”分数最高,代表其最接近于用户

想要查询的位置,具体分数可在标志 3 的悬浮框中查看,为“0.7453846153846154”,在地图上的位置在标志 2 处,仔细观察可知,其确实在广八路与八一路相交的路口附近。

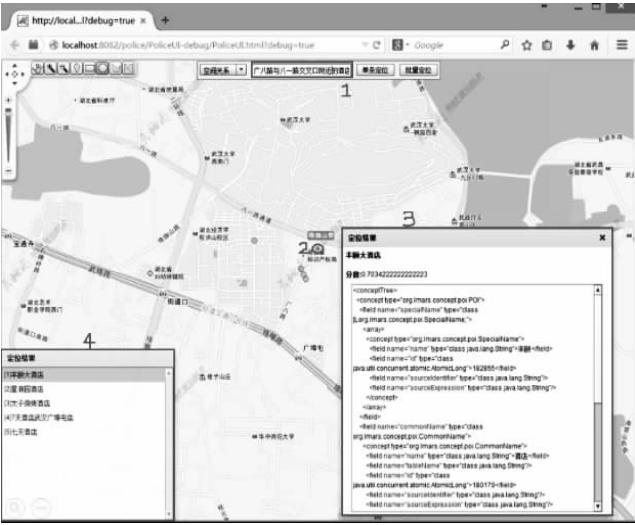


图 7 原型系统界面

相较于其他传统的搜索平台,如百度、Google,此平台能较好地处理包含空间关系的复杂 POI 查询。本文列举了几个具有不同特点的复杂 POI 描述,分别在百度、Google 和本平台上进行实验比较,实验结果如表 1 所示。

表 1 实验统计结果

No.	实验例子	百度	Google	本文
1	武汉广八路与八一路交叉口的酒店	错误	无	正确
2	武汉建设大道与长江日报路口上岛咖啡店	无	正确	正确
3	武汉和平大道建设一路路口处的酒店	错误	无	正确
4	武汉广八路与八一路交叉口附近的医院	错误	无	正确
5	武汉君益宾馆对面的桑拿	无	错误	正确
6	武汉亚贸对面的医院	错误	无	正确
7	武汉广埠屯附近的如家	正确	正确	正确
8	武汉雅阁酒轩附近的学校	正确	无	正确
9	武汉玫瑰南街附近的幼儿园	正确	无	正确
10	武汉四海酒楼附近的派出所	正确	无	正确

表 1 中“错误”代表查询结果不为零,但是不正确;“无”代表查询结果为零;“正确”代表查询结果正确。实验中,由于数据的缺乏,本文列举了 10 个比较有特点的例子进行实验比较并对结果进行分析。例 1~3 中只包含有简单的相交空间关系,其中,例 1 和 3 比例 2 更加模糊,例 2 的描述中出现了要查询的 POI 的名称的特名(SpecialName),即未登录词,是区分于其他 POI 名称的词汇,对于例 2,Google 和本文都查询正确,对于例 1 和例 3 的含有相交空间关系的模糊查询,百度和本文都查询正确;例 4 中包含了多个空间关系(邻近空间关系和相交空间关系),对于这种复杂描述,百度和 Google 都未能给出正确的查询结果,本文查询结果正确;例 5 和例 6 包含比较复杂的方位距离关系,即在计算“对面”的时候要考虑距离和方位特征,对于此种情况,百度和 Google 均不能正确查询,本文查询结果正确;例 7 至例 10 均只包含简单的邻近空间关系,且结果显示,百度对于这种的模糊查询的效果比较好,但 Google 的结果不好,本文的方法能正确查询。从以上分析来看,对于含有空间关系描述的中文 POI 的模糊查询(即不含特名),百度和本文的效果比较好,其中,百度对于含有邻近空间关系的 POI 模糊查询的效果较好,但对于较复杂的空间关系,如:相交空间关系

或多种空间关系,百度的查询结果没有本文的方法好。
结果表明,本文的 POI 地理编码方法比传统的方法能更好地解析匹配复杂的 POI 位置描述短语,并能较准确地查询到接近于用户描述的 POI。

4 结束语

本文通过分析武汉市大量的自然语言位置描述案例,根据自然语言语义位置描述特点对 POI 及空间关系进行建模;针对传统地理编码方法中的不足,基于一种融合语义本体的对象级规则匹配方法,提出了将空间计算融于 POI 地理编码的过程中,对于包含空间陈述的复杂 POI 位置描述短语,成功实现了 POI 的准确查询,并对查询结果集进行打分排序;介绍了原型系统和实验结果,并将此原型系统与百度、Google 相比,成功证明了此方法能较好地解析匹配复杂的 POI 位置描述短语,并能较准确地查询到接近于用户描述的 POI。下一步的研究工作将集中于查询包含其他自然语言空间关系或更加复杂的自然语言空间关系的 POI 名称的计算查询,并进一步优化计算查询中的打分机制,力求让查询结果集中最符合用户描述的查询结果反馈给用户,同时,将搜集大量的实验例子进行实验分析比较。

参考文献:

[1] 陈旭.地理编码引擎的设计与实现[D].辽宁:辽宁工程技术大学,2009.

[2] 姚盛江,吴洁,李树坤,等.地理编码技术及其在 PGIS 中的实现[J].警察技术,2011(3):14-17.

[3] 张林曼,吴升.地理编码系统中地名地址分词算法研究[J].测绘科学,2010,35(2):46-48.

[4] 徐飞,孙劲光.中文分词切分技术研究[J].计算机工程与科学,2008,30(5):126-128.

[5] 刘件,魏程.中文分词算法研究[J].微计算机应用,2008,29(8):11-16.

[6] 颜维龙,盖杰,武港山,等.面向网络的全文检索中索引文件的组织[J].计算机应用研究,2002,19(11):124-126.

[7] Atlam E S, Ghada E M, Fuketa M, et al. A compact memory space of dynamic full-text search using Bi-gram index [C]//Proc of IEEE Symposium on Computers and Communications. [S. l.]:IEEE Press, 2004:104-109.

[8] 龙军.基于角色标注的中文 POI 名称匹配的研究及原型系统实现[D].重庆:西南大学,2008.

[9] 梁明,罗荣,胡最.基于 Lucene 和 PostGIS 的地图搜索研究[J].测绘通报,2014(11):42-45.

[10] 杜冲,司望利,许璐.基于地理语义的空间关系查询和推理[J].地理信息科学学报,2010,12(1):48-55.

[11] 袁焯城,刘海江,裴韬,等.基于语义知识的空间关系识别研究[J].地球信息科学,2014,16(5):681-690.

[12] 张雪英,张春菊,杜超利.空间关系词汇与地理实体要素类型的语义约束关系构建方法[J].武汉大学学报,2012,37(11):1266-1270.

[13] 杜清远,任福.空间信息的自然语言表达模型[J].武汉大学学报,2014,39(6):682-688.

[14] 周春辉,朱欣焰,苏科华,等.基于 LBS 的兴趣点查询与更新机制研究[J].微计算机信息,2009,25(7-1):143-145.

[15] 杜世宏,王桥,李治江. GIS 中自然语言空间关系定义[J].武汉大学学报,2005,30(6):533-538.