基于浅层语义分析技术的语义检索

孙志军 郑 烇 袁 婧 刘 恒 王 嵩
(中国科学技术大学自动化系 合肥 230026)

摘 要 在信息检索领域,语义检索技术较传统的关键字检索,无论在检索效果还是用户体验方面,都有诸多优势。语义检索融合了信息检索、语义分析以及信息融合等诸多方法,已成为现阶段该领域研究的一项重要技术。在Lucene索引技术基础之上,提出了语义检索的方法,即对语句进行语义分析,获得一种描述语句浅层语义信息的形式化表示,并对这种形式化表示建立索引;将表述语义联系的多层次相似度通过信息融合技术进行融合,并将其映射成查询语句与索引数据之间的相似度,达到语义检索的目的。

关键词 语义检索,相似度融合,索引技术,语义分析,Lucene
中图法分类号 TP391.3 **文献标识码** A

Semantic Retrieval Based on Shallow Semantic Analysis Technology

SUN Zhi-jun ZHENG Quan YUAN Jing LIU Heng WANG Song
(Department of Automation, University of Science and Technology of China, Hefei 230026, China)

Abstract In the field of information retrieval, semantic retrieval performs better than traditional keywords retrieval in many aspects including effectiveness and User Experience. Semantic retrieval that has become an important technique in the specific domain integrates several methods such as information retrieval, semantic parsing and information integration. This semantic retrieval method based on Lucene analyzes sentences semantically and acquires a formal representation describing sentences' simple semantic content subsequently. Afterwards, index of this formal representation will be established. Since multi-levels' similarities demonstrating semantic relationships are merged by information fusion technology and mapped as similarities between query sentences and index data, the purpose of semantic retrieval will be achieved consequently.

Keywords Semantic retrieval, Similarity fusion, Indexing technology, Semantic analyzing, Lucene

1 引言

现今比较实用的信息检索系统主要采用关键词匹配技术。但是随着用户对检索效果的要求,关键词匹配技术在信息的语义和语用的揭示上存在局限性,这为语义检索的提出提供了条件。在语义检索的研究中,主流思想是基于本体理论[1-3],通过本体的语义网络和推理机制来实现概念间的语义关联。由于本体理论的特殊性,其强大健全的语义网络是本体实际应用的先决条件,也成为了制约其发展的重要因素。其他的一些研究从提取语义信息入手,通过挖掘词形背后的语义,来探索基于语义的概念匹配方法,从而提高检索系统的语义处理能力[4-6]。

本文意在借助汉语语句的语法和语义分析方法以提取语句的语义信息,采用信息检索和信息融合技术,实现基于汉语语句层面的语义检索。本文主要从语句的关键字、同义词和语义片段等多个层次,采用信息融合技术计算语句层面的相似度;同时在开源搜索引擎 Apache Lucene[7] 基础上加以改进,从而达到语义检索的要求。

2 总体描述

现阶段,通过语义分析从现代汉语语句中提取浅层语义信息,并以此作为语义检索的数据源的研究并不多见。语义分析技术大部分用于文本信息的过滤[8]、文本主题信息提取[9]等应用中。随着语义分析技术的不断发展和完善,我们发现将语义分析研究积累的成果用于语义检索的研究中,可以有效地满足用户的需求。本文通过将用户查询语句与被查询语句在语义层面的相似度表示为用户查询意图与查询数据之间的语义联系,从而达到语义检索的目的;所提思想很好地避开了本体理论需要建立纷繁复杂的语义网络以及实现和计算复杂等诸多问题。

本文提出的总体思想及处理流程如图1所示。

对于待处理的文本信息,通过语义分析的方法提取出其中的语义信息,并对语义信息建立索引,以备查询需要。对于用户的查询数据,采用同样的方法处理成可供查询使用的检

到稿日期:2011-07-01 返修日期:2011-09-28 本文受国家高技术研究发展计划(863)(2008AA01Z147),科技部支撑计划(2008BAH28B04),国家发改委CNGI课题(CNGI-09-03-14)资助。

孙志军(1986-),男,硕士生,主要研究方向为网络传播系统与控制,E-mail:myemailadnol@163.com;郑 烇(1970-),男,博士,副教授,主要研究方向为网络传播系统与控制、网络多媒体、媒体内容分发;袁 婧(1988-),女,硕士生,主要研究方向为网络传播系统与控制;刘 恒(1986-),男,硕士生,主要研究方向为网络传播系统与控制;王 嵩(1975-),男,讲师,主要研究方向为计算机网络、媒体内容分发。

索信息。其中，索引信息是以关键字和语义片段的形式存储的，下面将较为详细地描述语义信息的具体形式和建立索引的方法。根据 Apache Lucene 检索技术能够计算出关键字、语义片段等索引信息的相似度，然后采用信息融合技术将计算出的各个语义层次的相似度进行融合，得到最终查询语句与系统文本数据的相似度，从而完成整个检索过程。其中信息融合的具体步骤将在下文中详细描述。

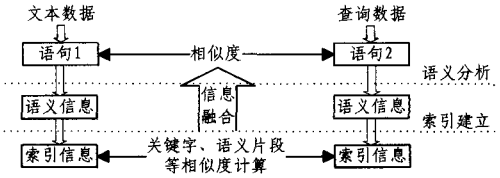

图 1　总体描述

3　算法实现

3.1　语义分析

语义分析一直是许多计算机从业者研究的重要课题，也是自然语言理解领域需要解决的根本性问题和追求的目标。语义分析是指在分析句子的句法结构和每个词词义的基础上，推出能够反映该句子意义的形式化表示。通过语义分析，可以理解人类自然语言，并进行深入的知识获取推理，从而抽取出自然语言语句背后的语义信息，使计算机与人类能无障碍地沟通。当然，让计算机能够理解人类的自然语言是很困难的，人们已经进行了多年的努力，虽然获得了一些成果，但总体效果并不理想。

Gildea 等人使用经验主义的方法进行语义角色标注[10](Semantic Role Labeling)的研究，这一研究领域被称为“浅层语义分析(Shallow Semantic Parsing)”，它是近几年来自然语言理解领域的重大突破。该方法并不对整个句子进行详细的语义分析，而只是将句子中的一些成分标注为给定动词的语义角色，这些成分被赋予一定的语义含义，并作为此动词框架的一部分。浅层分析摒弃了深层成分和关系的复杂性，因而能在真实语料环境下实现快速分析算法，获得比深层分析(full parsing)更高的正确率。

3.1.1　语义分析处理流程

本文借鉴前人的研究成果，通过语义分析完成了从现代汉语语句中提取浅层语义信息的功能。由于现代汉语语句的特殊性，借助汉语句型的研究成果在原有浅层语义分析技术的基础之上增加了汉语句型识别等分析过程，进一步提高了语义分析的效果。图 2 详细地描述了整个语义信息提取的过程，主要包括语法处理、语句过滤、语句主干提取、句型识别、修饰词语义提取和语义信息生成 6 个过程。

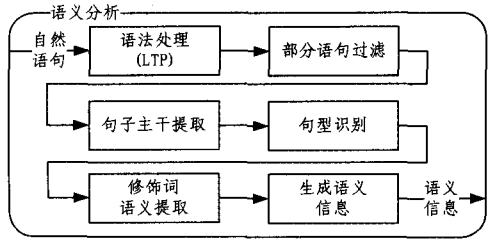

图 2　语义分析总体流程

1)语法处理：本过程使用由哈尔滨工业大学信息检索研究中心(HIT CIR)提供的语言技术平台(Language Technology Platform)[11]对自然语言语句进行预处理。其主要功能是对自然语言语句进行切词、词性标注和确定词之间的依存关系，即词之间的语法关系，如主谓关系(SBV)、定中关系(ATT)、动宾关系(VOB)等。例如对语句“我爱北京天安门”进行处理后得到如图 3 所示的结果。图 3 中，词下方用字母标注的表示词的词性，上方带箭头的曲线表示词之间的依存关系。

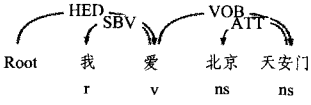

图 3　LTP 句法分析事例

2)语句过滤：因为以后流程需要的原因，对于语法处理结果中一些没有语义信息(如独词句)或者无法正常获取语义信息的语句，本模块会将其过滤掉。

3)语句主干提取：根据句法分析后，获取词之间的依存关系，去掉句中的修饰词，提取语句的主干。如在句子中出现定中关系、状中关系等即可去掉其中表示定语、状语的词。例如“我悄悄地走了，正如我悄悄地来”，其中“悄悄地”即为状语成分，需要舍弃。

4)句型识别：根据获取的句子的主干成分，通过分析主干成分的词性来确定该句子所属的句型。如当某一句子成分为“代词＋动词＋名词”时，即可确定该句子为主谓宾形式的陈述句；当句子成分为“名词＋动词＋名词＋动词”时，即可确定该句子为兼语句形式的陈述句。当然，汉语语句的句型还有很多，如连动、‘被’字句、‘把’字句、复合句型等。通过句型的识别，即可确定语句中概念之间存在的语义关系，如被字句中“A 被 B 打了”，即可确定其中的语义关系为“施事(B,打)”，“受事(打,A)”。

5)修饰词语义提取：在提取语义信息之前，需要分析句中的修饰词，提取其中可能存在的语义信息，如定语、补语等。

6)语义信息生成：根据前面多个过程的处理结果，分析并提取其中的语义信息，以某种形式表示出来。本文采用语义片段的形式，例“施事_A_B”、“时间_C_D”、“主谓宾_A_B_C”等。根据具体语义片段的长度，将语义片段分为 1 级、2 级、多级语义片段等。

经过上述流程，可以将自然语言语句处理为富含语义信息的语义片段，其中语义片段主要描述概念之间的语义关系。本文确定便于分析和记录的语义关系，如“施事”、“受事”、“时间”、“地点”、“拥有”等语义关系。通过这些标签来描述概念之间的语义关系，从而达到表示语句语义信息的目的。例如，对于语句“去年，我毕业了，独自居住在上海”，其处理结果为：“施事_我_毕业”，“施事_我_居住”，“时间_毕业_去年”，“地点_居住_上海”。

3.1.2　基于 LR 分析法的句型识别

本系统提出了基于现代汉语语句句型识别方法的浅层语义分析。现代汉语句式比英语复杂，所包含的句型也很繁多，如陈述句、疑问句、感叹句、祈使句、倒装句、连动句、兼语句、复合句、独词句等多种句式。系统针对其中较为常见的几种句式进行处理并加以识别。

在语义分析总体流程中，句型识别的主要步骤如下：

a)确定需要识别的句型，并构建分析器；

b)根据上一步骤“句子主干提取”过程获取句子主干词性内容；

c)基于构建的分析器分析句子主干词性内容，识别出句子的句型；

d)根据识别的句型，确定句子中的语义实体以及语义实体之间的关系。

句型识别分析器是整个句型识别流程中的核心部分，系统采用 LR(Left-Right)分析法构建分析器。LR 分析法采用从左到右扫描字符和最左规约过程，其归约过程是规范推导的逆过程，所以 LR 分析过程是一种规范归约过程。LR 分析器总体框架如图 4 所示。

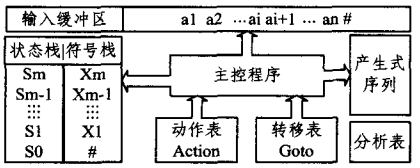

图 4　LR 分析器框架图

LR 分析器由 3 部分组成：

a)LR 主控程序：也称为驱动程序，对所有的 LR 分析器总控程序都是相同的。

b)分析表或分析函数：不同的文法分析表不同；同一个文法采用的 LR 分析器不同时，分析表也不同。分析表又可分为动作表(ACTION)和转移表(GOTO)两个部分，它们都可用二维数组表示。

c)分析栈：包括文法符号栈和相应的状态栈，它们均是先进栈、后出栈。

系统构建 LR 分析器的具体流程如图 5 所示。

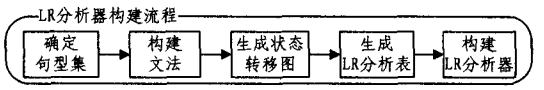

图 5　句型分析器构建流程图

通过以上流程，系统完成句型识别分析器的构建工作。首先，确定系统能够识别的句型集合，如陈述句、连动句、倒装句、兼语句等。再根据确定的句型集合构建语法分析文法。最后，基于 LR(Left-Right)自底向上分析法，将构建的文法生成相应的状态转移图，并生成对应的分析表，最终将分析表载入分析器中，构建 LR 分析器。

3.2　索引建立与检索

Lucene 是 Apache 软件基金会 jakarta 项目组的一个子项目，是一个高性能、可扩展的开源全文索引与检索引擎工具库。和绝大多数全文索引技术一样，Lucene 采用倒排索引技术[12]，能够对任意的、可转换为文本格式的数据进行快速的索引和搜索。它提供了一套简单且十分强大的核心 API 方便用户使用。

本文在基于 Lucene 全文检索技术的基础之上提出几点改进，从而实现语义检索的目的。

1)建立以语句为单位的索引

在传统的全文索引技术中，一般以文档为单位进行索引。本文提出了以语句为单位的索引。将文档根据具体内容划分为若干语句并建立索引；检索时，实际计算查询项与具体语句之间的相似度，再根据返回的语句映射到具体的文档。

2)将语句语义分析后获得的语义片段作为索引项

在全文索引技术中，通过对文本进行“切词”获得索引项，索引项是文本中出现的关键字。本文在原有基础上添加语义分析获得了语义片段，加强了语义信息对整个检索结果的影响。

3)加权设置描述不同语义信息的索引项

Lucene 支持用户对索引项的自定义加权，用户根据自身的需要，可以设置部分索引内容的权重。借助 Lucene 的此项功能对语句中不同语义层次的语义信息进行加权设置。具体索引项在语句中表达的语义强度不同，其相应的索引权重也不相同。

4)支持同义词扩展查询

为体现基于语义的检索，采用了支持同义词扩展查询的技术，使得查询效果更加完善。查询时，根据同义词与原词之间的相似距离设置查询权重。具体方法这里不细述了。

通过以上几点改进，实现了基于语句层面的语义检索。其中，具体的语句间相似度计算以及检索的加权设置等问题将在下面详细说明。

3.3　相似度融合技术

现在几乎所有的倒排索引技术均先计算查询项与索引项之间的相似度，再将其映射到查询语句与文档之间的相似度。本文在传统倒排索引技术基础之上，采用信息融合技术融合多层次的相似度，计算出查询语句与索引语句之间的相似度。

3.3.1　查询模型

常见的查询模型有以下两种。

1)向量模型

向量空间模型(Vector Space Model，VSM)是近年来应用较多的信息检索模型之一，这个模型对于查询与文档的相关度有较强的可计算性和可操作性。在向量空间模型中，文档和查询语句均被看成由索引项构成的向量，文档与查询语句之间的相似度用两个向量之间的夹角余弦表示。向量空间模型满足 Term 独立性假设：Term 在文档中的出现是独立的、互不影响的。

相似度计算方法：假设有 n 个索引项，查询 q 和文档 d 的向量表示分别为 $q=(a_1,a_2,\cdots,a_n)$ 和 $d=(b_1,b_2,\cdots,b_n)$，其中，a_i 表示第 i 个索引项在查询语句 q 中的权重，b_i 表示第 i 个索引项在文档 d 中的权重($i=1,2,\cdots,n$)。则相似度计算公式为

$$\mathrm{Sim}(d,q)=\frac{d\cdot q}{\|d\|\times\|q\|}=\frac{\sum_i(a_i\times b_i)}{\sqrt{\sum_i a_i^2\times\sum_i b_i^2}} \tag{1}$$

2)布尔模型

布尔模型是基于集合理论和布尔代数的一种简单的检索模型。由于集合的概念非常直观，因此布尔模型为信息检索系统的普通用户提供了一种易于掌握的框架。此外，查询被表示成有确切语义的布尔表达式。在布尔模型中，查询词 q 由连接词 not，and，or 连接起来的多个标引词所组成。这样，查询 q 本质上是一个常规的布尔表达式，它可以表示为多个向量的析取，即析取范式 DNF。

3.3.2　Lucene 评分公式

Lucene 在查询词汇的逻辑设置上采用了布尔模型，在评估查询语句与文档之间的相似程度上采用了向量模型。其给出的计算查询语句 q 和文档 d 的评分标准见式(2)[13]。其中

· $coord(q,d)$：协调因子，其值基于文档中包含查询的项

的个数，包含越多，权重越大。

• *queryNorm*(*q*)：标准化函数，所有文档的评分都会乘以这个标准化值，其不影响查询的最终排序，一般采用每个查询项权重平方和的倒数表示。

• *norm*(*t*,*d*)：域的标准化值，即在某一域中所有项的个数，通常在索引时计算该值，并将其存储到索引中。

• *tf*(*t_in_q*)：用于描述索引项的 *t* 在文档 *d* 中出现的频率，称为索引项频率(term frequency)，即词频，表示索引项对描述文档内容的能力。

• *idf*(*t*)：*df* 表示全部文档集 *D* 中出现索引项 *t* 的文档数(document frequency)；*idfk* 为 *dfk* 倒数，称为逆文档频率(inverted document frequency)，表示索引项区分其所在文档和其它文档的能力。

• *t. getBoost*()：表示索引项查询时添加的权重，这里权重的设置由用户自行调节。

$$\mathrm{Sim}(d,q)=coord(q,d)\cdot queryNorm(q)\cdot\sum_{t\_in\_q}(tf(t\_in\_q)\cdot idf(t)\cdot t.getBoost()\cdot norm(t,d)) \quad (2)$$

3.3.3 基于 Lucene 的相似度融合

对 Lucene 提供的评分机制加以分析，除去协调因子、标准化因子。对于任意一个索引项 *t*，其整个最终评分的贡献为 $tf*idf$ 再乘以索引项设置的权重 *t. getBoost*()。当然，这个可以看作是该索引项与文档之间的相似度。如果先不考虑权重，则相似度计算公式为

$$\mathrm{Sim}(d,t)=tf(t\_in\_d)*idf(t) \quad (3)$$

通过以上索引项与文档之间的相似度计算公式，可以获得每个索引项的相似度。如上述，语句经语义分析后获得若干个关键字、1 级语义片段和 2 级语义片段。这里为简单起见，假设不存在更高级别的语义片段。查询语句所有索引项可分为关键字集合 SEM0、1 级语义片段集合 SEM1 和 2 级语义片段集合 SEM2。我们给出这样的融合多层次语义相似度的理论评分式(4)。

$$\mathrm{Sim}(d,q)=a\cdot(\theta_0\cdot\sum_{t\_in\_SEM_0}\mathrm{sim}(d,t)+\theta_1\cdot\sum_{t\_in\_SEM_1}\mathrm{sim}(d,t)+\theta_2\cdot\sum_{t\_in\_SEM_2}\mathrm{sim}(d,t)) \quad (4)$$

式中，*a* 为协调因子，对应 Lucene 评分公式中的 *coord*(*q*,*d*)、*queryNorm*(*q*)和 *norm*(*t*,*d*)的乘积。θ_0、θ_1 和 θ_2 分别表示为关键字、1 级语义片段和 2 级语义片段设置的不同权重，对应 Lucene 评分公式中的 *t. getBoost*()。

式(4)实现了将关键字以及语义片段表示的多个层次语义联系的相似度融合为查询语句与文档之间的相似度，其可以在 Lucene 的评分式(2)中得到具体的体现。由于本文提出的语义检索思想是基于 Lucene 实现的，因此该公式的具体实施可以通过调节 Lucene 评分公式中的 *t. getBoost*()来完成，以实现多层次语义相似度的融合。

4 实验分析

本文实验通过计算全文本检索即关键字检索和语义检索的查准率和查全率进行比较。查准率(精度)是衡量某一检索系统的信号噪声比的一种指标；查全率(召回率)是衡量某一检索系统从文献集合中检出相关文献成功度的一项指标。计算公式如下：

查准率＝(检出相关文档量/检出文档总量)×100%

查全率＝(检出相关文档量/系统内相关文档总量)×100%

分别测试现代汉语语句的多个常用句式列举用例，计算在使用不同句式进行查询时，全文本检索和语义检索的查准率和查全率。采用的句式有陈述句、连动句、倒装句、表语句和复合句，具体测试结果如表 1 所列。

表 1 实验结果对比表

计算	检索类型	陈述句	连动句	倒装句	表语句	复合句
查准率	全文本检索	54.8%	51.3%	66.7%	63.1%	63.0%
	语义检索	83.7%	89.0%	94.8%	73.6%	61.1%
查全率	全文本检索	67.5%	51.1%	57.7%	44.2%	43.8%
	语义检索	80.4%	77.9%	76.0%	64.2%	55.8%

为直观显示数据的对比情况，图 6 和图 7 给出查准率和查全率的比较。

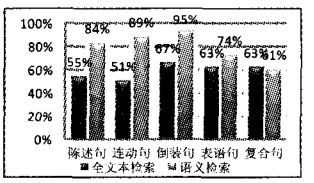

图 6 查准率对比图

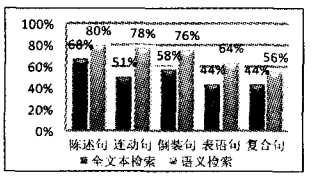

图 7 查全率对比图

通过上述实验结果可见，本文提出的语义检索思想较全文本检索在查全率方面效果有所提高；在查准率方面比全文本检索有较大的优势，尤其是在对陈述句、连动句、倒装句和表语句的查询中体现出很大的优势。但是其对于复合句查询效果仍不是很好，主要是因为语义分析对复合句的处理能力较弱。

总体而言，本文提出的思想对于简单句式的语义检索，其效果较传统的全文本检索有很大的提高。由于网络上索引的大部分文本还是以简单的句式为主，因此本文提出的方法在实际应用中仍能表现出较好的性能。

结束语 本文主要从语义分析和相似度融合技术的角度提出了一种语义检索的思想。在成熟的全文本检索 Lucene 基础上加以扩展，使之能够对语句语义处理的结果进行索引和检索；同时通过对 Lucene 的评分公式进行分析，并在此基础上加以改进，从而达到对语句多层次语义信息的相似度进行相似度融合的目的。通过实验分析，认为本文提出的关于语义检索的思想在处理现代汉语中绝大多数简单句式时有较好的效果。

参 考 文 献

[1] Arpirez J, Perez AG, Lozano A, et al. (onto)2agent: An Ontology-based WWW Broker to Select Ontology[C]// Gomez P A, Benjamins V R, eds. Proceedings of the Workshop on Application of Ontologies and Problem Solving Method. UK, 1998: 16-24

[2] Ontobroker [OL]. http://ontobroker. aifb. uni-karlsruhe. De/, 2011-02-14

[3] SKC[OL]. http://infolab. stanford. edu/SKC/, 2011-02-14

[4] Schubert F, Li Hui. Chinese word segmentation and its effect on information retrieval[J]. Information Processing and Man-agement, 2004, 40(1): 161-190

[5] Fu Guo-hong, Kit C, Webster J J. Chinese word segmentation as morpheme-based lexical chunking[J]. Dr-marionScfences, 2008, 178(9): 2282-2296

(下转第 146 页)

begin
for VL(Q)中与 p 相交的每个 VL(q)所对应的生成线段 q do
if $p \cap q \neq \Phi$ then
$R=R \cup \{(p,q),0\}$;
else 计算 d(p,q);
对于 $r \in R$ 且 r. $d \neq 0$ do
if $d_{min} < r.d$ then // d_{min}为最小距离//
$R=R-r$;
$R=R \cup \{(p,q),d_{min}\}$;
重构线段 p 局部 Voronoi 图;
end

3.3.2 删除线段的情况

一个线段集中删除了一条线段，若删除线段是结果集中的线段，且该线段与另一个线段集中的其他线段相交，则删除相交线段对即可；否则需要找到新的最近对。

算法 3 DELETE-VLCP(P,Q)(删除线段的更新算法)

输入：线段集 $P=\{p_1,p_2,\cdots,p_n\}$，$Q=\{q_1,q_2,\cdots,q_m\}$，线段集 P 中删除的线段 p；

输出：更新的最近对 R。

for $r \in R$ 且 r. $d=0$ do
if $p \in r$ then
$R=R-r$;
for $r \in R$ 且 r. $d \neq 0$ do
if $p \in r$ then
$R=R-r$;
找到算法 1 中 C 的 d 值次小的对象(u,v)；
$R=R \cup \{(u,v),d_{min}\}$;// d_{min}为当前的最小距离//
end

4 实验结果

实验是在 PentiumIV 2.8 GHz CPU，512MB 内存，Windows XP 平台上用 Visual C++ 6.0 实现的。

实验将本文算法与线性扫描算法(记为 DLCP)和 K-CPQ ($K=1$)进行比较。DLCP 依次计算一个线段集中每条线段与另一个线段集中每条线段之间的距离，并从中选取距离最小的线段对。

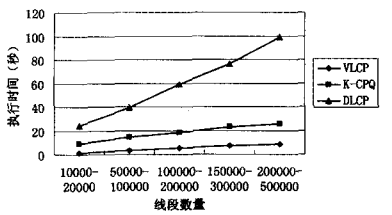

图 3 算法 VLCP 和算法 DLCP 及 K-CPQ 的比较

随机生成两个线段集，其线段数量分别为 10000，20000、50000，100000、100000，200000、150000，300000、200000，500000，实验比较 3 个算法的执行时间。图 3 给出了算法 VLCP 和算法 DLCP 及 K-CPQ 的执行时间的比较。由图 3 可知，算法 VLCP 的执行时间最短，算法 DLCP 的执行时间最长。这是因为随着线段数量的增加，算法 VLCP 通过 Voronoi 图划分区域，去掉了大量的候选，减少了比较次数，节省了大量时间。

结束语 本文给出了基于平面线段的最近对查询的定义，提出了解决该查询的方法和相关的定理及其证明。利用 Voronoi 图的邻接特性和局域动态特性确定候选，去掉了大量不可能成为候选的数据，减少了大量的距离计算。对数据集中增加线段和删除线段的情况分别进行了处理。通过实验验证了算法具有较好的性能。

参 考 文 献

[1] Corral A, Manolopoulos Y, Theodoridis Y, et al. Closest pair queries in spatial databases [C]// Proc. of 2000 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2000: 189-200

[2] Corral A, Manolopoulos Y, Theodoridis Y, et al. Algorithms for processing k-closest-pair queries in spatial databases [J]. Data and Knowledge Engineering, 2004, 49(1): 67-104

[3] Yang C, Lin K-I. An index structure for improving closest pairs and related join queries in spatial databases[C]// Proc. of the International Database Engineering and Applications Symposium. Piscataway: IEEE, 2002: 140-149

[4] 徐红波，郝忠孝. 一种基于 Z 曲线近似 k-最近对查询算法[J]. 研究计算机研究与发展，2008，45(2)：310-317

[5] Liu X, Liu Y, Xiao Y. Processing constrained K closest pairs query in spatial databases[J]. Wuhan University Journal of Natural Sciences, 2006, 11(3): 543-546

[6] Bespamyatnikh S, Snoeyink J. Queries with segments in voronoi diagrams [C]// Proc. of the 10th Annual ACM-SIAM Symp on Discrete Algorithms. Baltimore: ACM SIAM, 1999: 122-129

[7] 郝忠孝，王玉东，何云斌. 空间数据库平面线段近邻查询问题[J]. 计算机研究与发展，2008，45(9)：1539-1545

[8] 杨泽雪，郝忠孝. 基于 Voronoi 图的线段反向最近邻查询[J]. 计算机工程，2011，37(16)：30-32

[9] 郝忠孝. 时空数据库查询与推理[M]. 北京：科学出版社，2010：119-120

[10] 周培德. 计算几何[M]. 北京：清华大学出版社，2005：235-237

[11] Kolahdouzanm, Shahab I C. Voronoi-based k nearest neighbor search for spatial net work databases[C]// Proc. of VLDB. Toronto, Canada: [s. n.], 2004

(上接第 110 页)

[6] Zhang Mao-yuan, Lu Zheng-ding, Zou Chun-yan. A Chinese word segmentation based on language situation in processing ambiguous words[J]. Information Sciences, 2004, 162(3/4): 275-285

[7] Apache Lucene[OL]. http://lucene.apache.org/, 2011-02-18

[8] 刘永丹，曾海泉，李荣陆，等. 基于语义分析的倾向性文本过滤[J]. 通信学报，2004，25(7)

[9] 赵佳鹤，王秀冲，刘亚欣. 基于语义分析的主题信息采集系统的设计与实现[J]. 计算机应用，2007，27(2)

[10] Gildea D, Jurafsky D. Automatic labeling of semantic roles[J]. Computer, Linguist, 2002, 28(3): 245-288

[11] LTP[OL]. http://ir.hit.edu.cn/ltp/, 2011

[12] 郑榕增，林世平. 基于 Lucene 的中文倒排索引技术的研究[J]. 计算机技术与发展，2010，20(3)

[13] 吴众欣，沈家力. Lucene 分析与应用[M]. 北京：机械工业出版社，2008