一种基于权重矩阵的临近词检索问题解决框架

乔亚男[1] 齐 勇[1] 史 楠[1] 侯 迪[1] 王 晓[2]
(西安交通大学电信学院计算机系 西安 710049)[1] (第四军医大学唐都医院 西安 710038)[2]

摘 要 传统的信息检索模型假设查询中的关键词之间是并列关系,但用户的需求往往应该被抽象为一系列的关键词组,组内的关键词间具有更为紧密的语义关系,这就是定义的临近词检索问题。提出了基于权重矩阵的临近词检索问题解决框架,该框架将文档和查询抽象化为文档的权重矩阵表示和查询权重矩阵,通过计算两个矩阵间的相似度来实现临近词检索。实验结果证明,针对临近词检索问题,传统的信息检索模型只是一种简化问题的解决方案,权重矩阵框架从理论上和形式上更加契合临近词检索问题,查准率得到了显著的提高。

关键词 信息检索,权重矩阵,向量空间模型

Weigh Matrix Based Solution Framework for Term Proximity Information Retrieval

QIAO Ya-nan[1] QI Yong[1] SHI Yi[1] HOU Di[1] WANG Xiao[2]
(Dept. of Computer Science & Engineering, Xi'an Jiaotong University, Xi'an 710049, China)[1]
(Tangdu Hospital, the Fourth Military Medical University, Xi'an 710038, China)[2]

Abstract Tradional information retrieval models assume that keywords in queries are parallel, but the requirements of users should be abstracted to a series of keywords groups, and the sematic relations of keywords inside the group are closer than outside. This is "Term Proximity Information Retrieval"(TPIR) defined in this paper, and we presented a solution framework based on Weigh Matrix(WMSF). WMSF abstractes documents and queries to Weigh Matrix Representation of Document and Query Weigh Matrix, and then implements the TPIR based on the caculating of similarity between them. Empirical results show that WMSF is appropriate for TPIR compared with traditional information retrieval models which simplify the TPIR problems actually.

Keywords Information retrieval, Weigh matrix, Vector space model

1 概述

信息检索模型是整个信息检索领域研究的重要组成部分之一,如何帮助用户从文档集中找出自己需要的文档,是信息检索模型要解决的关键问题。用户如何定义"需要的文档"呢?给出感兴趣的关键词,是一种通用的解决方案。Google的成功为基于关键词信息检索的技术和市场奠定了基础。

但是,目前常用的搜索引擎和几种经典的信息检索模型,包括向量空间模型[1-2]、概率模型[3-5]、统计语言模型[6,7]和它们的一些衍生模型,在对待查询中的关键词时,都有一个很容易被人忽视的假设:关键词和关键词之间是并列关系(最多是有序的并列关系,排在前面的关键词权重比较大)。比如用户想在搜索引擎中检索有关感冒的症状以及吃什么药可以治疗感冒的网页,可能会输入"感冒 AND 症状 AND 药物"。看似正确,也能部分检索出你需要的网页。但这3个关键词在含义上很明显不是并列关系,"症状"和"药物"是从属于"感冒"的。具体地说,用户其实需要这样一种网页,在关键词"感冒"的周围出现了"症状"以及"药物"。如果一个网页尽管出现了"感冒"和"症状"(或"感冒"和"药物"),但两者之间却相距甚远,那很可能只是一篇文章中两个没有语义关系的孤立的词,并不满足用户的实际需求。严格地说,上面这个需求的查询表达式应当是"(感冒 PROX 症状) AND(感冒 PROX 药物)"(操作符"PROX"代表"临近")。本文把包含临近关系(Proximity)的信息检索问题称为临近词检索问题。

针对用户查询中关键词内部存在的语义关系问题,国际上已有一些研究者给出了自己的解决方案,如 Tao[8], Metzler[9], Beigbeder[10], Petkova[11] 和 Rasolofo[12]。这些研究的不足之处在于只给出了局部性问题的解决,没有对临近词检索问题在总体上做一个概括性的定义,影响了问题解决方案的通用性。无疑,现有的信息检索模型无法直接处理临近词检索问题。但临近词检索问题在形式上是一般信息检索问题的一种直接推广,应当也可以用现有信息检索模型的某种推广以处理。针对4个经典模型:布尔模型、向量空间模型、概率模型和统计语言模型,研究者们提出了很多改进和衍生模型。布尔模型和向量空间模型在实践中的应用非常广泛,而理论研究的热点在于向量空间模型、概率模型和语言模型。

到稿日期:2009-01-16 返修日期:2009-03-04 本文受863基金项目(2006AA01Z101),教育部博士点基金(20060698018)和陕西省科技攻关项目(2006K04-G23)资助。

乔亚男 博士研究生,主要研究方向为信息检索与服务组合;齐 勇 教授,博士生导师,主要研究方向为分布式系统与中间件技术;侯 迪 副教授,主要研究方向为数据库与中间件技术。

可以看出，向量空间模型是4个经典模型中唯一实践和理论研究都比较活跃的模型，这是由向量空间模型自身的特点决定的。首先，它所基于的代数理论比布尔模型基于的集合论在处理信息检索问题时要精细，相应模型的功能更强，可深挖的研究点也多；其次，代数理论中的相关数学计算要比概率模型基于的概率论和语言模型基于的随机过程中的数学计算要更适合计算机处理，相关算法的效率较高，便于投入实际的工程应用。换句话说，向量空间模型在这4种模型中拥有最合适的粒度，在功能和效率之间找到了一个平衡点。因此，笔者认为，应当对向量空间模型进行某种推广来尝试解决临近词检索问题。

向量空间模型将文档和查询抽象为文档向量和查询向量，将信息检索的过程视为文档向量和查询向量进行的相似度匹配的过程，是一维的；而临近词检索问题的关键点是词与词之间的语义关系，是二维的，因此我们可以设想将文档和查询抽象为一维向量的二维推广——矩阵，并以此为基础给出临近词检索问题的一般解决方案。

本文提出了一种基于权重矩阵的临近词检索问题解决框架，该框架将文档和查询抽象化为文档的权重矩阵表示和查询权重矩阵，通过计算两个矩阵间的相似度函数来实现临近词检索。实验结果证明，针对临近词检索问题，传统模型只是一种简化问题的解决方案，权重矩阵框架从理论上和形式上更加契合临近词检索问题，相比传统模型提高了查准率。

本文首先在第2节系统地定义了临近词检索问题，在第3节概述了权重矩阵框架的一些理论基础；在第4节为权重矩阵框架给出整体的阐述；第5节是实验；最后给出了结论和进一步工作的计划。

2 临近词检索问题

基于第1节的相关讨论，我们给出临近词检索问题的完整定义。

定义1(临近词检索问题) 一个信息检索问题给出的 k 个关键词 $t_1,t_2,t_3,\cdots,t_k$，如果可以分为 $m(m\leqslant k)$ 个关键词组，每组内部的关键词和组间的关键词相比，具有相对更紧密的语义关系，隐含要求组内的关键词在命中的文档中距离临近，则该问题称为临近词检索问题。

例如以下这组关键词：

$$\overbrace{\{t_1,t_2\}}^{\text{组内}},\{t_3,t_4\},\underbrace{\{t_5,t_6\}}_{\text{组间}},\cdots,\{t_{k-1},t_k\}$$

t_1,t_2 为同组的关键词，之间的语义关系比不同组的 t_4,t_5 更为紧密。

临近词检索问题可以分为下面几种类型：

- 当 $m=k$ 时，即每个关键词均独立成组，该临近词检索问题则退化为一般信息检索问题，即没有临近词关系的信息检索问题。如“感冒 AND 腹泻”；
- 当 $m=1$ 时，即所有的关键词均处于一个组，此时的临近词检索问题称为纯临近词检索问题。如“感冒 PROX 症状”；
- 当 $1<m<k$，关键词分组至少2组，且至少有一组组内的关键词大于等于2个时，称此时的临近词检索问题称为混临近词检索问题。如“(感冒 PROX 症状) AND(腹泻 PROX 药物)”。

如果一组中关键词数目大于两个，则需要通过对用户实际需求的分析将其拆分为若干个子组，每组内只有两个关键词。如查询“感冒 PROX 症状 PROX 药物”，该关键词组的完全拆分应为“(感冒 PROX 症状) AND(感冒 PROX 药物) AND(症状 PROX 药物)”，但从常理分析，用户的本意应为“(感冒 PROX 症状) AND(感冒 PROX 药物)”。

上面的几种临近词检索问题中关键词组与组织间的逻辑关系均为 AND，当需要使用 OR 操作符时，OR 操作符连接的两个子表达式可视为两个独立的临近词检索问题。如“(感冒 PROX 症状) OR (腹泻 PROX 药物)”，可视为两个独立的纯临近词检索问题“(感冒 PROX 症状)”和“(腹泻 PROX 药物)”的复合。

3 理论基础

3.1 基本概念

文本的内容特征常常用它所含有的基本语言单位(字，词，词组，或短语等)来表示，这些基本的语言单位被统称为文本的项(Term)，即文本可以用项集(Term List)表示为 $D(t_1,t_2,\cdots,t_n)$，其中 t_k 是项，$1\leqslant k\leqslant n$。项集中的项是可重复和有序的。项集类似于文档的一个摘要，项的顺序和文档中的顺序一致。

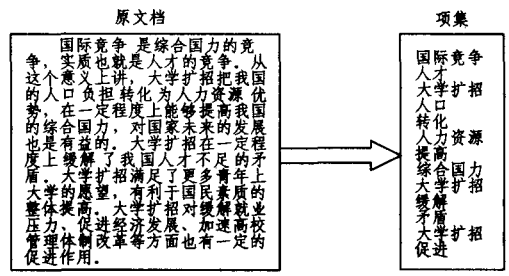

图1 原文档抽象为项集

如图1所示，从左侧的原文档中抽取出的项形成右侧的项集。项不一定是一个词，可能是一个词组、短语，乃至一个概念(concept)。项相当于文档的抽样，但并不一定是文档的“关键”词，也就是说，我们可能根据信息检索的需求，选取对整个文档来说并不重要的词(短语、概念等等)作为项。对于不同的需求，文档所抽取出的项集可能相距甚远。

本文采用的一些关于项的基本概念定义如下：

定义2(同类项) 从信息需求的角度看概念相同的项互称为同类项。比如“大学扩招”和“高校扩招”就属于同类项。

定义3(项向量和项元素) 项集中经过同类项合并后剩余的项组成的向量称为项向量。项向量不再具有同类项。为了不和项集中的项混淆，项向量中的项称为项元素。一个项向量中的项元素是不可重复且无序的。

定义4(项对) 文档中任意两个项的组合可以表示为 $P(t_i,t_j)$。

定义5(同类项对) 若干个项对 $P_1,P_2,\cdots,P_n$，如果这些项对中相对应的项都是同类项，则它们互称为同类项对。

3.2 临近词相关度分布

临近词检索问题的核心是“临近词”，如何准确地衡量文档中临近词之间的语义相关度呢？从语言学常理来讲，在一篇文档中距离较近的两个词的语义关系应当较强，距离相对较远的两个词的语义关系也应当较弱，随着距离的加大，语义

关系逐渐减弱，直至为 0。由此可见两个词之间语义关系随着距离的变化产生的变化应当是一个连续的过程，但如何描述这个连续的过程目前仍有很多种观点，各自有各自的适用范围。一种简单直观的想法是设定一个阈值 w，两个词之间的距离小于等于 w 就认为它们属于临近词，否则就不属于。该方法虽然过于粗粒度，临近词间的语义相关度非 0 即 1，但计算速度很快，w 值设置合适的话应用效果也不差，适用于重视速度的应用场合；粒度比较合适的观点认为临近词之间的语义相关度和距离成反比，距离越近语义相关度越大，反之则越小。该想法符合前面所说的语言学常理，但距离和相关度之间具体的函数关系仍需要确定。下面我们引入相关度分布函数的概念来准确地量化临近词之间的语义相关度。

定义 6(相关度分布函数)　描述临近词语义相关度分布状态的函数。对于项 t_1 和项 t_2，相关度分布函数通常可以表示为：

$R(t_1,t_2)=K(d(t_1,t_2),w)$，其中 $0\leqslant R\leqslant 1$

$R(t_1,t_2)$为语义相关度，w 为项 t_1 和项 t_2 距离的最大值，大于这个值就认为两个项不是临近词；$d(t_1,t_2)$是 t_1 和 t_2 之间的距离；K 我们称之为临近词核函数，定义距离和相关度之间具体的函数关系。

特别地，当项 t_1 和项 t_2 为同一个项的时候，我们定义

$$R(t_1,t_1)=1 \tag{1}$$

对于项对 $P(t_1,t_2)$，项 t_1 和项 t_2 的语义相关度 $R(t_1,t_2)$又称为项对 $P(t_1,t_2)$的权重，可记作 $W_{P(t_1,t_2)}$。

临近词核函数在本文中一般定义为计算量较小的线性递减函数：

$$K_l(d(t_1,t_2),w)=1-\frac{d(t_1,t_2)}{w}$$

临近词核函数的定义可以借鉴词汇共现模型的很多研究结果。常见的词汇共现模型有多项式递减模型[13]、指数递减模型[14]与吸引和排斥模型(指数模型的变形)[15]以及项场模型[16]等。除了项场模型之外，这些改进模型都属于“递减模型”的范畴，认为项对的相关度随着两个项的距离的增加而减小，而在项场模型中，两个项距离比较小的时候相关度基本不随着距离的改变而改变，只有当距离大于一个临界值的时候，相关度才随着距离的增加而减小。

4　临近词检索问题解决框架

4.1　文档的权重矩阵表示

以项对的权重这个概念为基础，我们可以定义文档的权重矩阵：

定义 7(文档的权重矩阵)　对于文档 C 的一个项集 $D(t_1,t_2,\cdots,t_n)$，构造 n 维矩阵 M_D：

$$M_D=\begin{pmatrix} W_{P(t_1,t_1)} & W_{P(t_1,t_2)} & \cdots & W_{P(t_1,t_n)} \\ W_{P(t_2,t_1)} & W_{P(t_2,t_2)} & \cdots & W_{P(t_2,t_n)} \\ \cdots & \cdots & \cdots & \cdots \\ W_{P(t_n,t_1)} & W_{P(t_n,t_2)} & \cdots & W_{P(t_n,t_n)} \end{pmatrix}$$

则称 M_D 是文档 C 在项集 D 下的权重矩阵。

由于项对 $P(t_i,t_j)$和项对 $P(t_j,t_i)$是等价的，则有

$W_{P(t_i,t_j)}=W_{P(t_j,t_i)}$

因此权重矩阵一定是对称矩阵。而且由式(1)可知，权重矩阵都是下面的形式：

$$M_D=\begin{pmatrix} 1 & W_{P(t_1,t_2)} & \cdots & W_{P(t_1,t_n)} \\ W_{P(t_1,t_2)} & 1 & \cdots & WP(t_2,t_n) \\ \cdots & \cdots & \cdots & \cdots \\ WP(t_1,t_n) & WP(t_2,t_n) & \cdots & 1 \end{pmatrix}$$

项集 $D(t_1,t_2,\cdots,t_n)$的各项中可能存在同类项，所以在它的权重矩阵中也可能存在着同类项对，并且这些同类项对一定是成行和成列出现的，这时，就需要进行同类项对合并。

例如，对于项集 $D(t_1,t_2,t_3)$，如果 t_1 和 t_3 是同类项。则对于权重矩阵：

$$M_D=\begin{pmatrix} W_{P(t_1,t_1)} & W_{P(t_1,t_2)} & W_{P(t_1,t_3)} \\ W_{P(t_2,t_1)} & W_{P(t_2,t_2)} & W_{P(t_2,t_3)} \\ W_{P(t_3,t_1)} & W_{P(t_3,t_2)} & W_{P(t_3,t_3)} \end{pmatrix}$$

经过同类项组合并(列 1+列 3，行 1+行 3)和矩阵降维，就有：

$$M_C=\begin{pmatrix} W_{P(t_1,t_1)}+W_{P(t_3,t_1)}+W_{P(t_1,t_3)}+W_{P(t_3,t_3)} & W_{P(t_1,t_2)}+W_{P(t_3,t_2)} \\ W_{P(t_2,t_1)}+W_{P(t_2,t_3)} & W_{P(t_2,t_2)} \end{pmatrix}$$

$$=\begin{pmatrix} 2+2W_{P(t_1,t_3)} & W_{P(t_1,t_2)}+W_{P(t_3,t_2)} \\ W_{P(t_2,t_1)}+W_{P(t_2,t_3)} & 1 \end{pmatrix}$$

要注意的是 t_1 和 t_3 虽然为同类项，但并不是同一个项，所以虽然 $W_{P(t_1,t_1)}=1$ 和 $W_{P(t_3,t_3)}=1$，但 $W_{P(t_1,t_3)}$却不一定等于 1，需要根据 t_1 和 t_3 在文档中的相对位置去计算。

我们使用符号 *comb* 来代表权重矩阵的同类项组合并，即：

$M_C=comb(M_D)$

此时的 M_C 就是文档 C 对于项集 $D(t_1,t_2,\cdots,t_n)$的权重矩阵表示。很明显，权重矩阵表示 M_C 的维数等于项集 $D(t_1,t_2,\cdots,t_n)$经同类项合并后的项向量中元素的个数。

4.2　查询权重矩阵的构造

在权重矩阵框架中，用户的信息需求被加工、转换为查询权重矩阵，并用与文档权重矩阵类似的表示形式表示，即

$$M_Q=\begin{pmatrix} W_{P(t_1,t_1)} & W_{P(t_1,t_2)} & \cdots & W_{P(t_1,t_m)} \\ W_{P(t_2,t_1)} & W_{P(t_2,t_2)} & \cdots & W_{P(t_2,t_m)} \\ \cdots & \cdots & \cdots & \cdots \\ W_{P(t_m,t_1)} & W_{P(t_m,t_2)} & \cdots & W_{P(t_m,t_m)} \end{pmatrix}$$

m 为整个文档集中项元素的总数。

为举例方便起见，我们假设文档集中只有 4 个项[t_1，t_2，t_3，t_4]，这 4 个项分别对应[感冒，腹泻，症状，药物]，则该文档集中文档的权重矩阵表示为：

$$M_C=\begin{pmatrix} W_{P(t_1,t_1)} & W_{P(t_1,t_2)} & W_{P(t_1,t_3)} & W_{P(t_1,t_4)} \\ W_{P(t_2,t_1)} & W_{P(t_2,t_2)} & W_{P(t_2,t_3)} & W_{P(t_2,t_4)} \\ W_{P(t_3,t_1)} & W_{P(t_3,t_2)} & W_{P(t_3,t_3)} & W_{P(t_3,t_4)} \\ W_{P(t_4,t_1)} & W_{P(t_4,t_2)} & W_{P(t_4,t_3)} & W_{P(t_4,t_4)} \end{pmatrix}$$

现在根据第 2 节中临近词检索问题的 3 个类别分别给出它们的查询权重矩阵。

对于一般信息检索问题，如查询“感冒 AND 腹泻”，相应的查询矩阵为

$$M_{Q_1}=\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

对于纯临近词检索问题，如查询“感冒 PROX 症状”，则相应的查询权重矩阵为：

$$M_{Q_2}=\begin{pmatrix}0&0&1&0\\0&0&0&0\\1&0&0&0\\0&0&0&0\end{pmatrix}$$

类似地，用户如果需要有关治疗腹泻的药物的文档，即查询“腹泻 PROX 药物”，则相应的查询权重矩阵为：

$$M_{Q_3}=\begin{pmatrix}0&0&0&0\\0&0&0&1\\0&0&0&0\\0&1&0&0\end{pmatrix}$$

如果用户需要同时满足这两种条件的文档，即混临近词检索问题“(感冒 PROX 症状) AND(腹泻 PROX 药物)”，则相应的查询权重矩阵为：

$$M_{Q_4}=\begin{pmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{pmatrix}$$

对于查询“感冒 PROX 症状 PROX 药物”，经需求分析后其查询可以转换为“(感冒 PROX 症状) AND(感冒 PROX 药物)”，则相应的查询权重矩阵为：

$$M_{Q_5}=\begin{pmatrix}0&0&1&1\\0&0&0&0\\1&0&0&0\\1&0&0&0\end{pmatrix}$$

很明显，和向量空间模型中查询向量的情况类似，相对于文档权重矩阵，查询权重矩阵通常是十分稀疏的。

4.3 文档的权重矩阵表示和查询权重矩阵的匹配

在权重矩阵框架中，我们使用矩阵的数量点积作为文档的权重矩阵表示与查询权重矩阵之间的相似度函数 $\mathrm{sim}(M_C, M_Q)$。具体的定义如下：

$$\mathrm{sim}(M_C,M_Q)=\sum_{i=1}^{m}\sum_{j=1}^{m}M_{C_{i,j}}\cdot M_{Q_{i,j}}$$

m 为整个文档集中项元素的总数，$M_{C_{i,j}}$ 和 $M_{Q_{i,j}}$ 分别指文档的权重矩阵表示与查询权重矩阵在 i 行 j 列处元素的值。

由于查询权重矩阵通常是十分稀疏的，并且文档的权重矩阵表示和查询权重矩阵均为对称矩阵，因此该相似度函数的计算复杂度和空间复杂度并不大。

5 实验

实验使用的数据集为 20 Newsgroups。它是一个信息检索领域非常流行的实验数据集，包含从 20 个新闻组收集来的大约 20000 篇文档，分为 20 个类别，包括软件、硬件、摩托车、运动、医药和政治等等。和其它常用的数据集相比，20 Newsgroups 比较接近 Internet 中实际的语言文字环境，包含着一定数量的垃圾信息。

实验的工作平台为 IBM Eclipse 3.2.0 和 Apache Lucene 2.2。Eclipse 是 IBM 公司推出的集成开发环境，通常用于 Java 开发，对 C 和 C++ 的开发也有很好的兼容性。Apache Lucene 是一个纯 Java 的高性能全文搜索引擎开发库，非常适合于跨平台全文搜索引擎的开发[17]。

我们使用的查询都是双关键词查询，两个关键词之间都有一定的语义关系，用来模拟临近词检索需求。首先使用传统的向量空间模型方法进行检索作为基准，然后再使用权重矩阵框架进行检索，两者的检索结果都保留评分靠前的前 10 篇文档，以人工判断是否和查询相关(人工判断时使用二元相关度：相关为 1 不相关为 0)，分别计算出两个模型的 Precision@10，实验结果如表 1 和表 2 所列。

表 1 向量空间模型的实验结果

关键词	返回文档数	命中文档数@10
Clinton, support	80	4
driver, download	11	8
Sun, mass	39	1
job, software	69	3
car, price	116	7
平均精确度：		0.460

表 2 权重矩阵框架的实验结果

关键词	返回文档数	命中文档数@10
Clinton, support	29	7
driver, download	4	4
Sun, mass	11	8
job, software	21	6
car, price	64	7
平均精确度：		0.640

由表 1 和表 2 可以看出，针对临近词检索问题，权重矩阵框架与传统的向量空间模型相比，在查准率上有着明显的性能提升。

结束语 本文提出了一个基于权重矩阵的临近词检索问题解决框架，该框架使用权重矩阵来表示文档和查询，可以进一步挖掘出文档内部隐含的信息。理论分析和实验结果表明，由于传统的信息检索模型在解决临近词检索问题时实质上忽略了临近词之间的语义关系，因此与它们相比权重矩阵框架更适合临近词检索问题。

不可否认的是，本文针对临近词检索问题的分析和讨论仍有很多缺陷和不足，主要体现在两个方面：其一是文档权重矩阵和文档权重矩阵的相似度算法，本文采取的是直接计算数量点积的策略。严格来说，这种方法只是一种妥协的方法，在计算相似度的时候忽略了权重矩阵中各个项对权重的位置信息，提出一个更合乎要求的相似度算法是下一步工作中的重点之一；其二是实验，由于临近词检索问题相关的测试集的缺乏，本文中的实验只是在一个比较小的范围内进行的，我们计划构造更大规模的临近词检索测试集，以获得更精确、更有说服力的实验结果。

参 考 文 献

[1] Salton G, Wong A, Yang C S. A vector space model for information retrieval[J]. Communications of the ACM, 1975, 18(11): 613-620

[2] Salton G, Buckley C. Term weighting approaches in automatic text retrieval[J]. Information Processing and Management, 1988, 24(5): 513-523

[3] Fuhr N. Probabilistic models in information retrieval[J]. The computer Journal, 1992, 35(3): 243-255

[4] Robertson S E, Rijsbergen C J V, Porter M F. Probabilistic models of indexing and searching[C]// SIGIR '80: Proceedings of the 3rd annual ACM conference on Research and development in information retrieval. Kent, UK: Butterworth & Co, 1980: 35-56

[5] Turtle H, Croft W B. Evaluation of an inference network-based retrieval model[J]. ACM Trans. Inf. Syst., 1991, 9(3): 187-222

[6] Lafferty J, Zhai Chengxiang. Document language models, query models, and risk minimization for information retrieval[C]// SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM, 2001: 111-119

[7] Lavrenko V, Croft W B. Relevance based language models[C]// IGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM, 2001: 120-127

[8] Tao T, Zhai C. An exploration of proximity measures in information retrieval[C]// SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM, 2007: 295-302

[9] Metzler D, Croft W B. A markov random field model for term dependencies// SIGIR'05: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM, 2005: 472-479

[10] Beigbeder M, Mercier A. An information retrieval model using the fuzzy proximity degree of term occurences[C]// SAC '05: Proceedings of the 2005 ACM symposium on applied computing. New York, NY, USA: ACM, 2005: 1018-1022

[11] Petkova D, Croft W B. Proximity-based document representation for named entity retrieval[C]// CIKM '07: Proceedings of the sixteenth ACM conference on conference on information and knowledge management. New York, NY, USA: ACM, 2007: 731-740

[12] Rasolofo Y, Savoy J. Term proximity scoring for keyword-based retrieval systems[C]// ECIR '2003: Proceedings 25th European Conference on IR Research. 2003: 207-218

[13] Lu Song, Bai Shuo. Quantitative Analysis of Context Field in Natural Language Processing[J]. Chinese Journal of Computers, 2001, 24(7): 742-747

[14] Gao J, Zhou M, Nie J Y, et al. Resolving Query Translation Ambiguity using a Decaying Co-occurrence Model and Syntactic Dependence Relations[C]// SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. New York, NY, USA: ACM, 2002: 183-190

[15] Guo Feng, Li Shao-zi, Zhou Chang-le, et al. Co-occurrence Word Retrieval Based On the Lexical Attraction and Repulsion Model [J]. Journal of Chinese Information Processing, 2004, 18(6): 16-22

[16] Qiao Ya-nan, Qi Yong, He Hui. The Research on Term Field Based Term Co-occurrence Model[C]// SKG '07: Proceedings of the Third International Conference on Semantics, Knowledge and Grid. Washington, DC, USA: IEEE Computer Society, 2001: 471-474

[17] Apache Lucene Homepage. http://lucene.apache.org/

(上接第 163 页)

a_{33}}，比 6 小的属性集合为{a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{31}, a_{32}, a_{33}}，两个集合取交为{a_{11}, a_{12}, a_{21}, a_{31}, a_{32}, a_{33}}，恰为概念({4,6}, {a_{11}, a_{12}, a_{21}, a_{31}, a_{32}, a_{33}})的内涵。反过来，求出分别比{a_{11}, a_{12}, a_{21}, a_{31}, a_{32}, a_{33}}大的对象集合，取交集即为概念的外延。

结束语 基于优势关系的信息系统在现实生活中有着广泛的应用。本文从形式概念分析的角度对其进行了讨论。首先给出了基于优势关系的形式背景的概念，并在其上建立了对象偏序集、属性偏序集以及对象-属性偏序集。进一步讨论了这些偏序集上的偏序关系，得出了一些有益的结论。在此基础上，提出了基于优势关系的概念格的定义及构建方法。这些结论进一步丰富了概念格的理论，对于研究基于优势关系的信息系统的规则提取提供了新的思路。

参 考 文 献

[1] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts[C]// Rival I, ed. Ordered Sets. Dordrecht: Reidel, 1982: 445-470

[2] Tonella P. Using a concept lattice of decomposition slices for program understanding and impact analysis[J]. IEEE Transactions on Software Engineering, 2003, 29(6): 495-509

[3] Valtchev P, Missaoui R, Godin R, et al. Generating frequent itemsets incrementally: two novel approaches based on galois lattice theory[J]. Journal Experimental Theoretical Artificial Intelligence, 2002, 14(2/3): 115-142

[4] 张文修，姚一豫，梁怡. 粗糙集与概念格[M]. 西安：西安交通大学出版社，2006

[5] 曲开社，翟岩慧. 偏序集、包含度与形式概念分析[J]. 计算机学报，2006，29(2)：219-226

[6] 曲开社，翟岩慧，梁吉业，等. 形式概念分析对粗糙集理论的表示及扩展[J]. 软件学报，2007，18(9)：214-218

[7] 曲开社，翟岩慧，李德玉，等. 信息系统同态的性质及上下近似的不变性[J]. 计算机科学，2005，32(12)：168-171

[8] 梁吉业，王俊红. 基于概念格的规则产生集挖掘算法[J]. 计算机研究与发展，2004，41(8)：1339-1344

[9] Kazimierz Z. Rough approximation of a preference relation by a multi-attribute dominance for deterministic, stochastic and fuzzy decision problems [J]. European Journal of Operational Research, 2004, 159: 196-206

[10] Salvatore G, Benedetto M, Roman S. Rough sets methodology for sorting problems in predence of multiple attributes and criteria[J]. European Journal of Operational Research, 2002, 138: 247-259.

[11] 徐伟华，张文修. 基于优势关系下协调近似空间[J]. 计算机科学，2005，32(9)：164-165

[12] 徐伟华，张文修. 基于优势关系下不协调目标信息系统的知识约简[J]. 计算机科学，2006，33(2)：182-184

[13] 张文修，仇国芳. 基于粗糙集的不确定决策[M]. 北京：清华大学出版社，2005

[14] Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations[M]. Berlin: Springer, 1999