ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.10, October 2006, pp.2096−2105
DOI: 10.1360/jos172096

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于最长顺序频繁词组的 Web 文献检索结构*

王大玲+, 于 戈, 鲍玉斌

(东北大学 信息科学与工程学院,辽宁 沈阳 110004)

A Web Bibliographies Retrieval Structure Based on the Longest Sequential Frequent Phrases

WANG Da-Ling+, YU Ge, BAO Yu-Bin

(School of Information Science and Engineering, Northeastern University, Shenyang 110004, China)

+ Corresponding author: Phn: +86-24-23895654, Fax +86-24-23895654, E-mail: dlwang@mail.neu.edu.cn, http://www.neu.edu.cn

Wang DL, Yu G, Bao YB. A Web bibliographies retrieval structure based on the longest sequential frequent phrases. *Journal of Software*, 2006,17(10):2096−2105. http://www.jos.org.cn/1000-9825/17/2096.htm

Abstract: Most Web bibliographies cannot meet the retrieval requirements of the researchers with different academic levels. The reason resulting in the problem is analyzed, and the idea of constructing an auxiliary Web bibliography retrieval structure for the users to obtain more proper bibliographies is proposed. Based on the idea, an algorithm of mining the longest sequential frequent phrases for extracting features of the bibliographies is designed, and an extended feature hierarchical tree describing the relationship among the features, among the bibliographies, and among the features, the bibliographies and its construction is presented. The experiments show that the new method outperforms the current popular TFIDF method in extraction features. The theoretical analysis explains that the extended feature hierarchical tree has constringent structure, reveals the relationship between phrases and bibliographies, and provides better assistant retrievals.

Key words: longest sequential frequent phrases; extended feature hierarchical tree; feature extraction; text mining; information retrieval

摘 要: 目前,大多数 Web 文献不能满足不同层次科研人员的查询要求.分析了这一问题产生的原因,提出建立辅助的 Web 文献检索结构以帮助用户更准确地获取所需文献的思想.基于该思想,设计了通过挖掘最长顺序频繁词组抽取文献特征的算法,提出了能够表现特征之间、文献之间、特征与文献之间关系的扩展的特征层次树结构及其构建方法.实验表明,挖掘最长顺序频繁词组在抽取文献特征方面比常用的 TFIDF 具有更大的优势.理论分析说明,扩展的特征层次树具有压缩的存储结构、词组与文献关系的表现方式和更好的辅助检索功能.

关键词: 最长顺序频繁词组;扩展的特征层次树;特征抽取;文本挖掘;信息检索
中图法分类号: TP311 文献标识码: A

科研人员在选择和确立一个新的研究课题时,常常需要上网查阅相关文献.从一篇文献中,他们能够领会文献的思想和方法,通过作者的主页了解作者及其从事的其他研究工作,通过参考文献追溯其所属领域的发展历

* Supported by the National Natural Science Foundation of China under Grant Nos.60573090, 60473073, 60503036 (国家自然科学基金); the Foundation for University Key Teacher by the Ministry of Education of China (国家教育部高等学校骨干教师资助计划)
Received 2005-05-05; Accepted 2005-12-13

史以及相关的会议或刊物等.然而,虽然有许多搜索引擎和专业网站为科研人员提供帮助,但却难以满足他们的上述要求,其原因包括科研人员对信息的搜索能力有限和信息的组织结构不完善两个方面.

一般地,科研人员查询文章的手段包括:① 输入文献的主题作为关键字直接查找;② 通过会议或期刊的相关主题来查找;③ 根据领域知识输入其他相关主题进行间接查找.然而,这些查询方法均未能够很好地解决问题.因为:① 现在的大多数搜索引擎均使用基于关键字的查询技术,即只有其标题包含输入的关键字的文章才能被找到,而实际上,许多与用户输入的关键字相关的文章,其标题中却不包含该关键字;② 如果科研人员选择一个新的研究课题,他们可能不了解相关的会议和期刊,而查找所有的会议论文和期刊显然是不可能的.而且,很多文章具有多个主题,但会议论文或期刊只能将一篇文章安排在一个主题中;③ 一个刚刚选择新研究课题的人,并不一定具有足够的相关知识,因此难以实现前述的"间接查询".

我们认为:建立一种合适的 Web 文献组织结构,为不同层次的科研人员提供辅助查询是十分重要的.这种合适的 Web 文献组织结构应尽可能完整地涵盖文献的信息,同时准确地表达文献之间的关系.因此本文一方面通过文本挖掘技术获取表现文献内容的特征;另一方面,根据这些特征建立一种联系特征和文献关系的 Web 文献检索结构.据此,本文的贡献为:① 定义最长顺序频繁词组及相关概念,分析最长顺序频繁词组对于表现文献特征的作用;② 提出一种最长顺序频繁词组挖掘算法,并通过实验证明最长顺序频繁词组表现文献特征的优势;③ 提出一种基于最长顺序频繁词组建立 Web 文献检索结构的算法,并讨论该 Web 文献检索结构的应用.

1 最长顺序频繁词组

1.1 相关定义

定义 **1**(最长顺序频繁词组 **LSFP**). 设 D 为一个文本,p 为 D 中的一个词组.如果 p 在 D 中出现的频度满足一个给定阈值,则称该词组为频繁词组.频繁词组所包含的词数称为频繁词组的长度.如果频繁词组的长度为 i,标识该词组为 i-phrase.对于任意 i-phrase(i>1),例如 i=2,如果认为由 $word_1word_2$ 组成的词组与由 $word_2word_1$ 组成的词组是不同意义的,那么称这种词组为顺序词组,称符合上述规则的频繁词组为顺序频繁词组,记为 SFP (sequential frequent phrase).令 *S-Set* 是 SFP 的集合,且 $p \in S\text{-}Set$,如果不存在 p' 满足 $p' \supset p$ 并且 $p' \in S\text{-}Set$,则称 p 为最长顺序频繁词组,记为 LSFP(longest sequential frequent phrase).

我们认为,LSFP 比 SFP 能够更准确地表述文章的意义.例如,"medical image mining"是一篇文章的 LSFP,而"image mining"是该文的 SFP,显然,前者在表述文章的意义方面比后者更为准确.

一般地,一篇文章的标题和关键字能够比较准确地描述该文的内容,但各网站中诸多的文献(如 Lecture Notes in Computer Science 以及由 IEEE Computer Society Press 出版的会议文集)均没有关键字项,而标题中有限的词不能全面地表述文献的内容,因此,从文中抽取关键字具有重要意义.我们根据"频繁出现的词更有意义"的直觉强调频繁词组,根据"不同的词序通常表示不同的意义"的直觉强调顺序频繁词组,根据"最长词组能够更准确地表述文献的意义"的直觉强调最长顺序频繁词组.因此,我们将从文献中抽取 LSFP,用以描述文献的特征.

1.2 LSFP挖掘算法

我们借鉴倒排索引(inverted index)技术[1],通过对一篇文献中的词建立其出现位置的倒排索引来进行 LSFP 的抽取.

倒排索引主要用于通过关键词从大批文档中检索出包含关键词的文档.每个文档都可以用一系列关键词来表示,这些关键词描述了文档的内容.只要找到文档,便可以找到文档中的关键词.反之,如果按关键词建立到文档的索引,便可以根据关键词快速地检索到相关文档.具体地,关键词被存储在索引文件(index file)中(比如按字母顺序存储),对于每个关键词,均有一个指针链表,表中的每个指针都指向出现过该关键词的文档,所有指针链表构成置入文件(posting file).

例如,一篇文档 d 的内容为 d={data mining algorithms are applied in data mining application},设频度阈值 *frequency*=2.经 stopword (http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/porter)和 stemming(http://

www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words)处理后,对 d 建立倒排索引的过程如图 1 所示.

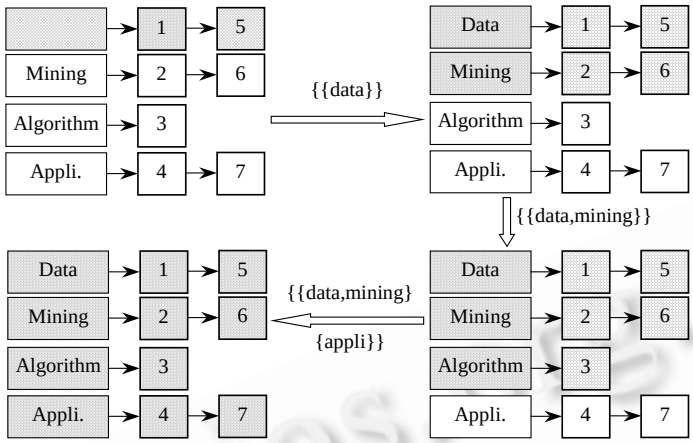

Fig.1 An example of inverted index for a text

图 1 应用倒排索引处理文档的实例

在此数据结构支持下,从文献中挖掘 LSFP 算法如下所示:

算法 **1**. 从文献中挖掘 LSFP.

输入:文本 d,用户输入的最小频度阈值 *frequency*;

输出:LSFP 的集合:*LSFP-Set*;

方法:

1) 对文本 d 进行预处理,得到词集 $T=\{t\}$;
2) 建立 T 的位置倒排索引$\{P_{t1},P_{t2},\ldots,P_{tm}\}$;
3) If $|P_t|<frequency$,从 T 中删除词 t;
4) For each $t\in d$
5) {$templist=P_t$;LSFP+=t;
6) While $|templist|>=frequency$
7) {If $|currentlist=comparelist(templist,P_{t\text{-}next})|>=frequency$
8) LSFP+=t_{next}; $templist=currentlist$;
9) Else break;
10) }
11) *LSFP-Set*+=LSFP;
12) }

其中,函数 $comparelist(templist,P_{t\text{-}next})$以当前词的位置链表 *templist* 和位置倒排索引中下一个词 t_{next} 的位置链表 $P_{t\text{-}next}$ 为参数,返回另一个位置链表 *currentlist*,该链表表示为 $currentlist=\{p|p\in P_{t\text{-}next}\wedge\exists p'\in list\wedge p-p'=1\}$,即链表 *currentlist* 内的位置索引均为在 $P_{t\text{-}next}$ 中出现过、且在 *templist* 中存在相应的较之小 1 的位置.如果 *currentlist* 中的位置索引的数目小于用户输入的最小频度阈值 *frequency*,那么可以判定,当前 LSFP 就是最长的序列频繁词组,则退出循环,继续寻找下一个 LSFP;如果 *currentlist* 中的位置索引的数目超过 *frequency*,则继续判断下一个词是否符合条件,直到找到最长顺序频繁词组为止.最后,还要还原用 Stemming 处理的词.

可以把找到的 LSFP 用于文本对象的向量表示之中,以便更好地表示文本的内容.同时,LSFP 的另一个优点就是它的选取可以仅针对单一的文本,而无须像许多特征选取方法那样必须对一批文本进行处理.

1.3 性能评价

1.3.1 数据集和评价准则

我们使用两个数据集进行特征抽取结果的比较,一个是 oai_citeseer 数据集(http://citeseer.IST.psu.edu/OAI/

html),该数据集包括约 100 000 篇包含作者、标题和摘要的文章,其中 440 篇还包含关键词.我们将这 440 篇文章作为测试集,记为 Dataset1.我们分别选择前 100,200,300,400,440 篇文章,从摘要中抽取特征,与原文中的关键词进行比较.另一个数据集是从计算机学术期刊《Data Mining and Knowledge Discovery》,《Machine Learning》和《World Wide Web: Internet and Web Information System》中随机选择的 160 篇文章的作者、标题、关键词和摘要作为测试集,记为 Dataset2.我们分别选择前 40,80,120,160 篇文章,从摘要中抽取特征,与原文中的关键词进行比较.

我们借鉴分类和聚类的评价方法,给出式(1)~式(3)所示的评价准则.

$$Recall=n_{KE}/n_K,\quad precision=n_{KE}/n_K \tag{1}$$

这里,n_K 为文本中存在的关键词数目;n_E 为从该文本中抽取的关键词数目;n_{KE} 为从该文本中正确抽取的关键词数目.在英文文献中,一个关键词可能包含多个词,因此,n_{KE} 的定义为:设 $keyword_1$ 是一篇文本中的关键词,由 n 个词组成,$keyword_2$ 是从该文本中抽取的关键词,表示为 $w_{21}w_{22}\ldots w_{2m}(m\le n)$.如果$\forall i(i=1,2,\ldots,m)$满足 $w_{2i}\subset keyword_1$ 且 $m=n$,则 n_{KE} 加 1;如果$\forall i(i=1,2,\ldots,l)$满足 $w_{2i}\subset keyword_1$ 且 $l<m$ 或 $l=m$ 且 $m<n$,则 n_{KE} 加 l/n.

据此定义,一篇文本 i 的 F 测度 F_i-$Measure$ 则如式(2)所定义.

$$F_i\text{-}Measure=2\times Recall\times precision/(Recall+precision) \tag{2}$$

一个数据集 D 的 F 测度 F_{total}-$Measure$ 的计算如式(3)所示.

$$F_{total}-Measure=\left(\sum_i F_i-Measure\right)\Big/|D| \tag{3}$$

1.3.2 实验方法

TFIDF[2]是当前一种比较流行的抽取文本特征的技术,它依据某个词的词频及其出现过的文本的频率来计算该词在整个文本集合中的权重,依据权重进行特征选取.权重越高,说明该词对文本的区分能力越强;否则,其区分能力则越弱.因此,我们将通过与 TFIDF 的比较来评价挖掘 LSFP 算法的性能.

我们应用 TFIDF 权重方法分别从 Dataset1 和 Dataset2 中抽取前 20,50,100,200,300,400 个词组作为关键词.同时,考虑到文本的标题与内容在表征文本意义方面所具有的不同的重要性,我们在应用 TFIDF 时将为在标题中出现的词赋予更大的权重.我们还分别设置频度阈值为 2,3 和 4,应用挖掘 LSFP 方法在 Dataset1 和 Dataset2 中抽取关键词.

我们将应用 TFIDF 在文本中抽取关键词、不考虑标题的方法称为“TFIDF”,将应用 TFIDF 在文本和标题中抽取关键词(即文本和标题中词的权重均赋为 1)的方法称为“TFIDF Title 1”,将 TFIDF 在文本中抽取关键词、赋予标题中词的权重为 2 的方法称为“TFIDF Title 2”,将应用挖掘 LSFP 从文本中抽取关键词的方法称为“LSFP”.我们将比较上述 4 种方法应用于 Dataset1 和 Dataset2 的 F_{total}-$Measure$.

1.3.3 结果与分析

图 2~图 5 分别是在 Dataset1 上应用上述 4 种方法测得的 F_{total}-$Measure$.图 6~图 9 分别是在 Dataset2 上应用上述 4 种方法测得的 F_{total}-$Measure$.

从图 2~图 4 可见:选择前 300 个关键词时,TFIDF,TFIDF Title 1 和 TFIDF Title 2 在 Dataset1 中均具有最大的 F_{total}-$Measure$;从图 5 又可见:当设置频度阈值为 3 时,LSFP 在 Dataset1 中具有最大的 F_{total}-$Measure$.

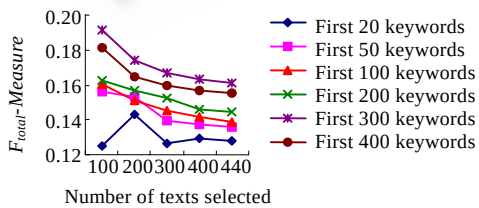

Fig.2 F_{total}-$Measure$ using TFIDF in Dataset1

图 2 在 Dataset1 上应用 TFIDF 的 F_{total}-$Measure$

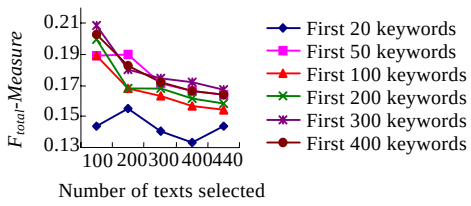

Fig.3 F_{total}-$Measure$ using TFIDF Title 1 in Dataset2

图 3 在 Dataset1 上应用 TFIDF Title 1 的 F_{total}-$Measure$

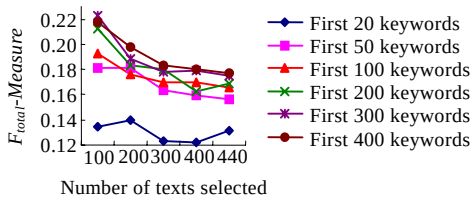

Fig.4 F_{total}-*Measure* using TFIDF Title 2 in Dataset1

图 4 在 Dataset1 上应用 TFIDF Title 2 的 F_{total}-*Measure*

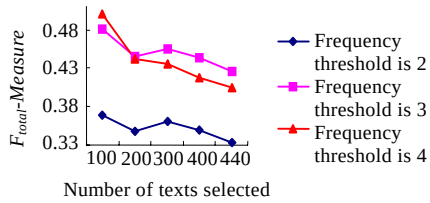

Fig.5 F_{total}-*Measure* using LSFP in Dataset1

图 5 在 Dataset1 上应用 LSFP 的 F_{total}-*Measure*

从图 6~图 8 可见:选择前 50 个关键词时,TFIDF,TFIDF Title 1 和 TFIDF Title 2 在 Dataset2 中均具有最大的 F_{total}-*Measure*;从图 9 又可见:当设置频度阈值为 4 时,LSFP 在 Dataset2 中具有最大的 F_{total}-*Measure*.

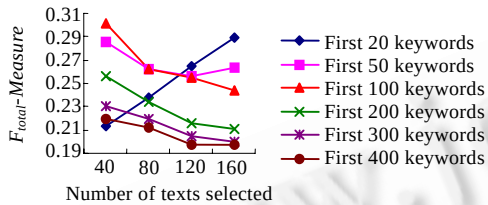

Fig.6 F_{total}-*Measure* using TFIDF in Dataset2

图 6 在 Dataset2 上应用 TFIDF 的 F_{total}-*Measure*

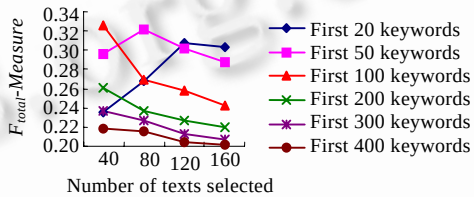

Fig.7 F_{total}-*Measure* using TFIDF Title 1 in Dataset2

图 7 在 Dataset2 上应用 TFIDF Title 1 的 F_{total}-*Measure*

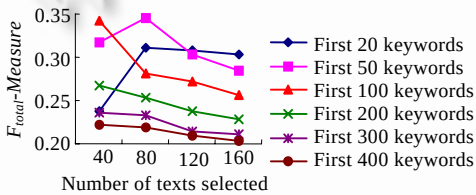

Fig.8 F_{total}-*Measure* using TFIDF Title 2 in Dataset2

图 8 在 Dataset2 上应用 TFIDF Title 2 的 F_{total}-*Measure*

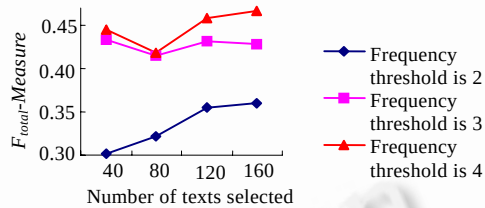

Fig.9 F_{total}-*Measure* using LSFP in Dataset2

图 9 在 Dataset2 上应用 LSFP 的 F_{total}-*Measure*

根据上述结果,我们选择前 300 个关键词并应用 TFIDF,TFIDF Title 1 和 TDIDF Title 2,设置频度阈值为 3 并应用 LSFP,在 Dataset1 上比较它们的 F_{total}-*Measure*,结果如图 10 所示.我们还选择前 50 个关键词并应用 TFIDF,TFIDF Title 1 和 TDIDF Title 2,设置频度阈值为 4,并应用 LSFP,在 Dataset2 上比较它们的 F_{total}-*Measure*,结果如图 11 所示.

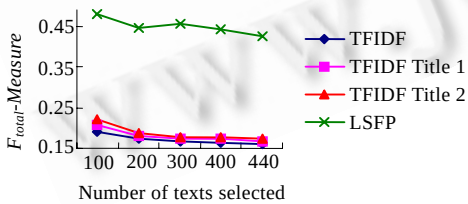

Fig.10 Comparison of different methods in Dataset1

图 10 在 Dataset1 上应用各种方法的比较

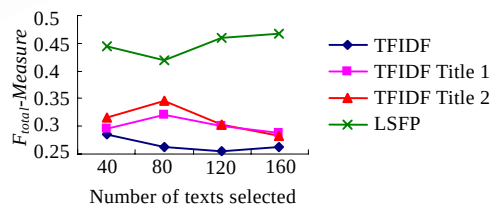

Fig.11 Comparison of different methods in Dataset2

图 11 在 Dataset2 上应用各种方法的比较

从图 10 和图 11 可见:① 无论在 Dataset1 还是 Dataset2,TFIDF Title 2 的 F_{total}-*Measure* 略大于 TFIDF Title 1,而 TFIDF Title 1 略大于 TFIDF.这是因为 TFIDF Title 1 考虑了标题对于表述文本的作用,而 TFIDF Title 2 则考虑了标题在表述文本意义方面较文本内容具有更大的作用.当然,图 10 显示:在 Dataset1 上,这种差异较小;而图 11 则显示:在 Dataset2 上,这种差异更大一些;② LSFP 方法在从文本中抽取关键词方面比 TFIDF 具有更大的优

势.图 10 显示:在 Dataset1 上,LSFP 具有明显的优势;而图 11 则显示:在 Dataset2 上,LSFP 虽然具有优势,但不如在 Dataset1 上那样明显.

实际上,TFIDF 在从文本集合中抽取特征或关键词方面起着重要作用,是目前比较流行的一种方法.一般地,集合中各文本的差异越大,应用 TFIDF 的效果越好.然而,如果集合中各文本差异较小,使用 TFIDF 权重方法抽取文本特征则不尽如人意.在 Dataset2 中,各文本分别来自于不同的计算机类学术期刊,差异较大,因而应用 TFIDF 的效果较 Dataset1 有一定优势.

进一步地,从图 2~图 5 可见:在 Dataset1 上,随着文本数量的增加,各种方法的 F_{total}-*Measure* 呈一定的变化规律.而从图 6~图 9 可见:在 Dataset2 上,各种方法的 F_{total}-*Measure* 没有明显的变化规律,我们认为,这是由于 Dataset2 中文本数量太少所致.

在科学研究领域,来自不同会议或期刊的文献可多可少,文献之间内容的差异或大或小,所以,采用 TFIDF 权重方法抽取文献特征并非总是有效的,即便考虑标题的作用也是如此.我们提出的通过挖掘 LSFP 抽取关键词的方法可以针对文献集合或单文献,也不受文献集合中各文献差异大小的限制,因此具有更广泛的应用空间.

2 Web 文献检索结构

我们从 Web 文献中抽取特征或关键词,旨在构建一个适合各层次查询的 Web 文献辅助检索结构.由前面分析可见,通过挖掘 LSFP 从文献中抽取特征具有更大的 recall 和 precision 的综合优势.因此,我们将基于 LSFP 构建一种能够表征文献-特征、特征-特征以及文献-文献之间关系的检索结构.为此,我们提出一种数据结构——扩展的特征层次树(extended feature hierarchical tree,简称 EFHT),并基于此构建 Web 文献检索结构.

2.1 扩展的特征层次树

假设从文献 D_1 中抽取的 LSFP 为 *AB*,*ACD* 和 *FA*(这里,*A*,*B*,*C* 和 *D* 是不同的词),从文献 D_2 中抽取的 LSFP 为 *A* 和 *CDE*,从 D_3 中抽取的 LSFP 为 *A*,*CD* 和 *F*.由于 *A* 与 *AB*,*CD* 与 *CDE* 来自不同的文献,因此,我们不仅可以将一篇文献与其 LSFP 建立联系,而且根据 $A\subset AB, CD\subset CDE$ 等关系,可以建立不同文献之间的联系.

定义 **2**(扩展的特征层次树 **EFHT**). 该树的根是一个空节点,每个中间节点表示一个 LSFP(简称 LSFP 节点),每个叶节点存储一个文献的名称(简称为文献节点).设一个 LSFP 节点存储的 LSFP 为 *lsfp*,其父节点存储的 LSFP 为 $lsfp_f$,其子节点(如果也是 LSFP 节点)存储的 LSFP 为 $lsfp_c$,则有 $lsfp_f\subset lsfp\subset lsfp_c$.设一个文献节点存储的文献名为 *D*,其所有的父节点存储的 LSFP 均是从 *D* 中抽取的 LSFP.由于这样的结构具有层次关系,又具有树的性质,但同时一个节点又具有多个父节点,并非通常意义的树结构,因此称其为扩展的特征层次树,记为 EFHT.

根据文献 D_1 包含的 LSFP 为 *AB*,*ACD* 和 *FA*、文献 D_2 包含的 LSFP 为 *A* 和 *CDE*、文献 D_3 包含的 LSFP 为 *A*,*CD* 和 *F* 的关系构建的 EFHT 如图 12 所示.EFHT 既表述了各 LSFP 之间的关系,又表达了 LSFP 与文献的关系.同时,通过各 LSFP 的关系,又将各文献建立起联系.

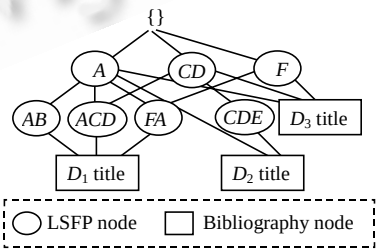

Fig.12 An example of EFHT

图 12 扩展的特征层次树实例

值得注意的是,在 EFHT 中,每个 LSFP 节点至少有一个子节点是文献节点,因为每个 LSFP 至少出现在一个文献中.同时,对于任一文献,如果 L_1 和 L_2 均为该文献的 LSFP,则不存在 $L_1\subset L_2$ 或 $L_1\supset L_2$,因为如果存在这种关

系,L_1 和 L_2 必有一个是非 LSFP.因此,任意 LSFP 节点不可能与其父、子节点具有相同的文献节点.然而,如果 L_1 和 L_2 来自两个不同的文献,则可以存在 $L_1 \subset L_2$ 或 $L_1 \supset L_2$,因为它们在各自的文献中可以均为 LSFP.

2.2 EFHT的构建

由图 12 的 EFHT 可见,如果采用该图所示的结构,则不仅存在大量的冗余,而且将导致其查询成为指数问题.因此,参考频繁模式存储结构 FP-Tree[3],我们提出 EFHT 的结构及基于此结构的构建和查询算法.

如图 12 所示的 EFHT 的存储结构如图 13 所示.该结构包括词头表以及由 LSFP 和文献标题构成的扩展树,扩展树中每个分枝代表一个 LSFP,分枝上每个节点是组成该 LSFP 的一个词,最终以该 LSFP 存在的文献节点结束;词头表中存储各 LSFP 的第 1 个词,指针指向该 LSFP 分枝的第 1 个节点.

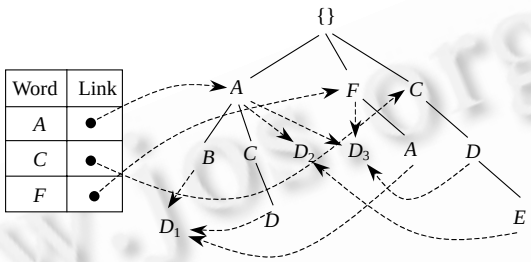

Fig.13 EFHT structure in Fig.12

图 13 图 12 所示的 EFHT 结构

针对此结构,我们给出 EFHT 的构建算法如下所示:

算法 **2**. 构建 EFHT.

输入:一文本集合 *D*;

输出:针对 *D* 的 EFHT;

方法:

```
1)  构建 EFHT 的空词头表 Word-Table 和根节点并赋值"{}";
2)  For all d∈D
3)   {将 d 读入主存;
4)    Call 算法 1 从 d 中挖掘 LSFP 集合 LSFP-Set;
5)    For every LSFP∈LSFP-Set
6)      {在 Word-Table 中查找 LSFP 的第 1 个词;
7)       If 找到
8)          {If 该节点对应分枝上包含该 LSFP {插入或链接 d 文献节点}
9)           Else {插入该 LSFP 各词节点,插入或链接 d 文献节点}
10)         }
11)      Else {在 Word-Table 中建立头词节点;
12)            在树中插入该 LSFP 各词节点,插入或链接 d 文献节点,与 Word-Table 链接;
13)            }
14)     }
15)  }
```

在算法的第 8)行、第 9)行和第 12)行,对于文本 *d*,在其第 1 个 LSFP 被插入 EFHT 或在 EFHT 中找到时,由于 EFHT 中尚不存在 *d* 的文献节点,所以需要插入;而 *d* 的其他 LSFP 被插入 EFHT 或在 EFHT 中找到时,仅将 LSFP 最后的词节点与 *d* 的文献节点链接即可.另外,我们将 Word-Table 按照 Word 项排序,从而减少了 EFHT 建立和检索时的查找时间.

2.3 EFHT的查询

对于任意一个文献 d,设 $lsfp$ 是来自 d 的任意一个 LSFP.根据图 13,在 EFHT 中,$lsfp$ 对应一个分枝 $branch$,$branch$ 上的每个中间节点都是组成 $lsfp$ 的一个词,叶节点是 d 的标题,$lsfp$ 中的第一个词出现在 Word-Table 中并与树中对应节点建立链接.同时,对应不同文献的 LSFP 还可能存在包含关系,表现在 EFHT 中同一分枝上的不同终点,如在图 13 中,CDE 结束于 D_2,CD 结束于 D_3,$CDE \supset CD$;F 结束于 D_3,FA 结束于 D_1,$FA \supset F$.进一步地,由于一个文献包含多个 LSFP,每个 LSFP 又与来自其他文献的 LSFP 具有包含关系,这样,EFHT 就构成了 LSFP 之间、文献之间以及 LSFP 与文献之间的层次关系.

根据这种关系,当用户输入一个查询词组 $phrase$ 时,首先根据 $phrase$ 中的第 1 个词、通过 Word-Table 找到 EFHT 中对应的分枝(如果存在),沿该分枝找到对应的文献标题.如果该分枝上除标题节点外还有尚未结束的子分枝,则这些子分枝便构成了 $phrase$ 的下层节点,可供用户进行进一步的查询.由于不同长度的 LSFP 可对应查询词组的不同深度,因此,可以满足不同层次用户的查询需求.

这样,当用户输入查询词组 $phrase$ 时,如果 $phrase$ 是来自任意文献的一个 LSFP,则用户的查询层次包括:

层 1:$phrase$ 对应的文献标题集合 D_1;

层 2:$phrase$ 的深层 LSFP 及所有来自 $d \in D_1$ 的 LSFP 的集合 L_1;

层 3:所有 $LSFP \in L_1$ 对应的文献标题集合 D_2(其中删除 D_1);

层 4:所有 $l \in L_1$ 的深层 LSFP 及所有来自 $d \in D_2$ 的 LSFP 的集合 L_2(其中删除 L_1);

层 5:所有 $LSFP \in L_2$ 对应的文献标题集合 D_3(其中删除 D_2);

层 6:所有 $l \in L_2$ 的深层 LSFP 及所有来自 $d \in D_3$ 的 LSFP 的集合 L_3(其中删除 L_2);

…

如果 $phrase$ 不是 LSFP,但能够通过 Word-Table 找到,则层 1 为 $phrase$ 的深层 LSFP 的集合,后面的层次依次类推.例如在图 13 中,如果用户输入的查询词组为 CD,则层 1 为 D_3(CD 对应的文献标题),层 2 为 CDE(CD 的深层 LSFP),A 和 F(来自 D_3 的 LSFP),层 3 为 D_2(A 对应的文献标题).如果输入的查询词为 AC,则层 1 为 ACD(AC 的深层 LSFP),层 2 为 D_1(ACD 对应的文献标题),层 3 为 AB 和 FA(来自 D_1 的 LSFP).在查询过程中,查询层次的深度将根据用户的需求和 EFHT 的构成随时终止或继续进行.

2.4 EFHT的评价

我们从以下 3 个方面对 EFHT 进行评价:① 结构上,EFHT 采用类似于 FP-Tree 的压缩存储结构,而 FP-Tree 在存储与查询方面的效率已被证明[3].另外,在 FP-Growth 算法中,为找到不同长度的频繁项集,不仅需要不断插入新节点,还要反复地对 FP-Tree 上各节点进行计数,同时还要根据不同的前缀生成条件 FP-Tree.在 EFHT 中,由于无须统计各节点(词)的计数,更无须产生条件 FP-Tree,而仅对其进行查询和插入操作,因此相关操作更加简单;② 内容上,EFHT 表达了文献与 LSFP,LSFP 与 LSFP 以及文献与文献之间的层次关系,而通过从文献中挖掘 LSFP 抽取文献特征的方法已在第 1.3 节中被证明是更有效的.因此,EFHT 的层次关系是文献特征之间的层次关系,关键词从短到长,并且长关键词包含短关键词,这样的关系反映在 EFHT 中就是任一分枝上从高层到低层的层次关系.从语法意义上看,长关键词的限定条件更多,因而所表征文献的内容更窄,也更深.这样,不同层次的关键词能够适应不同层次用户的查询需求;③ 应用方面,EFHT 提供了一种能够满足不同层次用户要求的辅助查询结构.事实上,我们并没有修改现有的搜索引擎,更没有建立新的搜索引擎.我们构建并应用 EFHT,仅仅是实现一个介于搜索引擎和用户之间的前端辅助搜索工具.当用户输入查询词时,根据 EFHT 的结构,一方面帮助用户找到相关的文献名称,另一方面引导用户进行更深入的或更相关的内容查询.最后,若真正找到所需的文献,还需借助于搜索引擎,只是用户的查询要求已在其最初输入的关键词的基础上,深度或广度发生了变化.

3 相关工作及讨论

在特征或关键词抽取方面,目前采用的主要方法是基于词在文献中出现的频度来抽取.文献[4]提出基于术

语频度、从文献的摘要和标题中抽取关键词的方法,文献[5]提出一种在单个文档中基于频繁词及与其“共现”词频度抽取关键词的方法.我们挖掘 LSFP 抽取关键词的方法与上述方法有类似之处,比如基于词组频度、在摘要和标题中抽取、仅考虑单个文献等.但我们的工作与上述工作的不同之处在于:首先,考虑最长频繁词组,这是基于“最长频繁词组能够更准确地表述文档的意义”的直觉;其次,考虑词组中各词出现的顺序,因为词的不同顺序常常代表不同的意义;第三,抽取过程不需要任何领域知识.当然,我们尚未完成从 Web 上自动获取相关文献的工作,而在此方面,文献[6]提出基于统计学和启发式规则的方法自动获取 Web 文献的方法,文献[7]提出应用支持向量机和隐 Markov 模型技术抽取文献重要元素的方法,这些是我们在进一步的工作中值得借鉴的.

在知识管理方面,采用概念层次管理方法最著名的是 WordNet(http://www.cogsci.princeton.edu/~wn/),它提供了一个词的同义词、反义词、同位语等,帮助用户获得一个术语的相关概念.然而,这里的相关概念是通用领域的,主要基于语法本身,一般不涉及专业领域的知识.例如,在数据库和人工智能领域中,“数据挖掘”与“知识发现”、“模式发现”等许多词是同义词,但 WordNet 中没有完全表现这种关系.文献[8]中提出两种技术 DISCERNER 和 EXTERNDER,以便自动地标识话题并相关于一个概念图.DISCERNER 将概念图组织成为一个索引帮助用户访问,EXTERNDER 描述概念图的话题并应用聚类技术组织话题.这两种方法与我们本文工作一样都包含知识获取和知识构建两部分内容.然而,该项工作中概念之间的联系形如“like”,“is a”,“has”等,而没有层次关系.文献[9]给出了两种基于频繁术语的文本聚类方法:FTC 用于单层次聚类,HFTC 用于多层次聚类.聚类的结果可将一个文本分为多个类,这如同本文工作中从一篇文献中抽取多个特征作为其标识一样.但是,文献[9]中的类层次来自于聚类的结果,而非特征(关键词)的关系,这是与我们工作的不同之处.文献[10]给出了建立术语间概念层次的方法,该方法在建立概念层次时,不仅根据术语本身的关系(这类似于我们的方法),也根据术语在意义上的关系,如“gas”和“battery”是“energy”的下层概念(在概念树中为子节点).然而,发现这样的关系显然需要借助于领域知识或本体.我们的方法是根据术语本身的包含关系建立层次,又可根据来自同一文献的多个特征以及包含同一特征的多个文献发现文献之间的关系,这种术语之间、文献之间、术语和文献之间的层次关系既可以辅助用户进行信息检索,又可以作为文本挖掘的一种初始的领域知识.

4 结束语

本文提出在文献中挖掘最长顺序频繁词组作为文献的特征,根据特征之间的层次关系建立扩展的特征层次树,根据特征与文献的关系推导出文献之间的关系,从而使用户在查询时根据上述关系、并借助于搜索引擎尽快获得所需的文献.同时,本文通过实验证明了所提出的挖掘最长顺序频繁词组的方法,在抽取文献特征方面比目前常用的 TFIDF 方法具有更大的优势.通过相关理论说明了扩展的特征层次树本身的特点和辅助用户查询方面的功能.因此,本文达到了如下目标:

① 最长顺序频繁词组是从文献中抽取出来的,即便它们不出现在文献的标题中,也同样可以被检索到;

② 扩展的特征层次树表达了特征之间、文献之间、特征与文献之间的关系,即便用户不具有相关知识,他们也能够获取到相关的文献信息;

③ 扩展的特征层次树中,特征之间的层次关系可以满足不同层次用户的查询需求.

然而,本文的研究还存在不足,改进这些不足是我们进一步的工作,具体为:

① 仅仅基于词或词组在文献中出现的频度抽取文献特征是不够的,因此,我们将考虑词义的关联性;

② 一篇文献中,不仅标题和内容能够提供特征信息,作者、出版机构及参考文献等同样能够提供关于该文献的特征信息,因此,我们将考虑文献的多维数据模型;

③ 本文的算法和数据结构本身也可以进一步优化,例如,考虑修改扩展特征层次树的结构,适用于 XML 格式,从而借助于该技术进行查询优化.

此外,本文的方法仅考虑了 Web 文献内容本身和用户的不同层次,而未考虑 Web 文献之间的链接关系和用户的个性化需求,未建立用户模型.因此,一些针对 Web 文献的链接分析技术[11]和基于用户 Profile 的信息检索模型[12]也是我们进一步工作中值得借鉴的.

References:

[1] Hong D, Kin K. Update conscious inverted indexes for XML queries in relational databases. In: Galindo F, Takizawa M, Traunm R, eds. Proc. of the 15th Int'l Conf. on Database and Expert Systems Applications. Berlin, Heidelberg: Springer-Verlag, 2004. 263−272.

[2] Kantrowitz M, Mohit B, Mittal V. Stemming and its effects on TFIDF ranking. In: Belkin N, Ingwersen P, Leong M, eds. Proc. of the 23rd Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2000. 357−359.

[3] Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In: Chen W, Naughton J, Bernstein P, eds. Proc. of the 2000 ACM SIGMOD Int'l Conf. on Management of Data. New York: ACM Press, 2000. 1−12.

[4] HaCohen-Kerner Y. Automatic extraction of keywords from abstracts. In: Palade V, Howlett RJ, Jain LC, eds. Proc. of the 7th Int'l Conf. on Knowledge-Based Intelligent Information and Engineering Systems. Berlin, Heidelberg: Springer-Verlag, 2003. 843−849.

[5] Matsuo Y, Ishizuka M. Keyword extraction from a single document using word co-occurrences statistical information. In: Russell V, Haller S, eds. Proc. of the 16th Int'l Florida Artificial Intelligence Research Society Conf. AAAI Press, 2003. 392−396.

[6] Geng J, Yang J. AutoBib: Automatic extraction of bibliographic information on the Web. In: Proc. of the 8th Int'l Database Engineering and Applications Symp. Los Alamitos: IEEE Computer Society, 2004. 193−204.

[7] Okada T, Takasu A, Adachi J. Bibliographic component extraction using support vector machines and hidden Markov models. In: Heery R, Lyon L, eds. Proc. of the 9th European Conf. on Research and Advanced Technology for Digital Libraries. Berlin, Heidelberg: Springer-Verlag, 2004. 501−512.

[8] Leake D, Maguitman A, Reichheraer T. Topic extraction and extension to support concept mapping. In: Russell V, Haller S, eds. Proc. of the 16th Int'l Florida Artificial Intelligence Research Society Conf. AAAI Press, 2003. 325−329.

[9] Beil F, Ester M, Xu X. Frequent term-based text clustering. In: Proc. of the 8th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM Press, 2002. 436−442.

[10] Lawrie D, Croft W, Rosenberg A. Finding topic words for hierarchical summarization (DRAFT). In: Croft W, Harper D, Kraft D, *et al*., eds. Proc. of the 24th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2001. 349−357.

[11] Xue GR, Yang Q, Zeng HJ, Yu Y, Chen Z. Exploiting the hierarchical structure for link analysis. In: Ricardo A, Ziviani N, Marchionini G, *et al*., eds. Proc. of the 28th Annual Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. New York: ACM Press, 2005. 186−193.

[12] Arezki R, Poncelet P, Dray G, Pearson DW. Information retrieval model based on user profile. In: Bussler C, Fensel D, eds. Proc. of the 11th Int'l Conf. on Artificial Intelligence: Methodology, Systems, and Applications. Berlin, Heidelberg: Springer-Verlag, 2004. 490−499.

王大玲(1962－),女,辽宁沈阳人,博士,教授,CCF 高级会员,主要研究领域为数据挖掘,Web 挖掘,文本挖掘.

于戈(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库理论与技术.

鲍玉斌(1968－),男,博士,副教授,CCF 高级会员,主要研究领域为数据仓库与 OLAP.