文章编号: 1003-0077(2008)03-0056-08

搜索引擎中的聚类浏览技术

李红梅[1,3],丁振国[1],周水生[2],周利华[1]

(1. 西安电子科技大学 计算机学院,陕西 西安 710071;
2. 西安电子科技大学 理学院,陕西 西安 710071;
3. 河北农业大学 信息科学与技术学院,河北 保定 071001)

摘 要: 搜索引擎大多以文档列表的形式将搜索结果显示给用户,随着 Web 文档数量的剧增,使得用户查找相关信息变得越来越困难,一种解决方法是对搜索结果进行聚类提高其可浏览性。搜索引擎的聚类浏览技术能使用户在更高的主题层次上查看搜索结果,方便地找到感兴趣的信息。本文介绍了搜索引擎的聚类浏览技术对聚类算法的基本要求及其分类方法,研究分析了主要聚类算法及其改进方法的特点,讨论了对聚类质量的评价,最后指出了聚类浏览技术的发展趋势。

关键词: 计算机应用;中文信息处理;搜索引擎;文档聚类;信息检索;聚类标识
中图分类号: TP391 **文献标识码**: A

Clustering Method of Web Search Results

LI Hong-mei[1,3], DING Zhen-guo[1], ZHOU Shui-sheng[2], ZHOU Li-hua[1]

(1. School of Computer Science and Technology, Xidian University, Xi'an, Shanxi 710071, China;
2. School of Science, Xidian University, Xi'an, Shanxi 710071, China;
3. College of Information Science and Technology, Agricultural University of Hebei, Baoding, Hebei 071001, China)

Abstract: Most search engines return ranked lists of document snippets, which makes the user difficult to find the relevant information. One method is that the snippets returned by the search engine are grouped into clusters, which may help the user quickly and efficiently navigate the results of a query at a topic level and locate the relevant information. This paper first introduces, several key requirements for Web research results clustering methods and the classification of the clustering methods. Then it probes into the major clustering algorithms and their improved method at present, and discusses the evaluation of clustering quality. Finally, this paper summarizes the future developments of clustering search engine results.

Key words: computer application; Chinese information processing; search engine; document clustering; information retrieval; cluster label

1 引言

随着信息技术迅速发展,互联网信息呈指数增长,人们越来越难以获取自己想要查找的信息,从 Web 这个异构、分布和动态的信息库中发现真正对用户有用的信息和知识是一个具有挑战性的课题。搜索引擎是资源发现的主要 Web 工具,已经成为人们查询获取信息的重要手段,但是目前大多数搜索引擎返回结果太多,且大多按照与用户查询的相关度进行排序,以庞大的文档列表形式显示给用户。事实上,相关度排序采用的标准并不能反映用户的

收稿日期: 2007-07-16 **定稿日期**: 2007-12-02
基金项目: 国家自然科学基金资助项目(60603098)

作者简介: 李红梅(1968—),女,博士生,讲师,主要研究方向为信息处理和人工智能;丁振国(1959—),男,博士,教授,主要研究方向为计算机网络与信息处理;周水生(1972—),男,博士,副教授,主要研究方向为优化算法及其理论、图像处理、模式识别。

查询意图,几乎一半的查询结果是与用户无关的[1],而对搜索引擎日志的分析则表明多数用户只愿意浏览 10～30 个查询结果[2],那么排列在后面的相关信息就很难被发现。另外,大多数查询趋向于短查询[3],由于查询词的多义性,使得查询结果往往包含多个主题内容,用户需要仔细浏览文档列表,排除不相关的内容,查找自己感兴趣的信息。因此,为了满足日益增长的网络用户对查询质量的要求,必须提高搜索引擎查询结果的可浏览性。

一种方法是采用 Web 文档分类技术[4～6],一般需要预先对分类器进行训练来建立整个 Web 分层类目,文献[6]利用了本体论的方法建立概念分层,然后将搜索结果映射到这些分层组织的类目中。这种分类方法过于复杂,不利于用户在不熟悉的领域查找新的主题;由于某些与查询相关的子主题并不存在于分类目录中,在组织特定查询的结果时也不是十分有效[7,8]。

针对上述问题,较好的解决办法是对搜索结果进行自动、非监督聚类。通过对搜索结果的内容进行聚类,创建类目体系,使同类中文档内容的相似度尽可能地大,而类与类之间文档的相似度尽可能地小,并对每个类目用相应的主题词加以描述。然后把类目呈现给用户,使用户能在更高的主题层次上来查看搜索引擎返回的结果,方便地查找到感兴趣的信息,从而可大大缩小用户所需浏览的结果数量,缩短用户查询所需要的时间,搜索结果的聚类浏览技术已经成为研究的一个热点。

2　聚类浏览技术的基本要求

大多数传统的聚类算法不能直接应用于搜索结果的在线聚类,其实用性对聚类算法提出了几个基本要求[9,10]:

(1) 相关性: 该算法应该能够聚类相同/相似的文档,把与用户查询条件相关的文档与不相关的文档分开。

(2) 概括性: 用户通过快速浏览就能找到自己感兴趣的内容,因此聚类算法需要对每个类目提供简明准确的概括描述,形成便于浏览的聚类标识。标识的质量取决于好的结构性(即文本符合句法和语法规范)、描述能力(即能够很好地描述聚类中所包含的内容)和区分能力(即能够很好地将所描述的类目与其他类目区分开来)[11]。

(3) 重叠性: 因为文档会涉及多个主题的信息,因此应该避免把每个文档只聚类到单独的一个类目,可以叠加聚类。

(4) 快速性: 聚类算法应该能够快速聚类,将查询结果显示给用户前不能有很大的延迟。

(5) Snippets 聚类: 由于搜索结果处理的实时性,大多数用户不愿等待系统下载原始文档形成聚类,因此,对搜索结果的聚类是基于短文文摘的,即 snippets 聚类,这就要求根据搜索引擎返回的标题和文摘(Snippets)也应形成高质量的聚类。snippet 聚类是区别检索结果聚类和一般文本聚类的主要指标之一。

因此,聚类的质量和速度与聚类算法的好坏有很大关系。

3　聚类浏览技术的分类

搜索引擎的聚类浏览技术实质上是为了方便用户的浏览,将聚类技术用于信息检索结果的可视化输出。聚类算法和聚类标识是聚类浏览技术的两个重要组成部分。聚类算法决定了搜索结果的组织结构和运行效率,而聚类标识则是帮助用户迅速确认生成的文档类目相关与否的重要信息[12],是提高可浏览性的基本体现。

聚类浏览技术按照聚类标识分为关键词标识(Single Words)和短语标识(Phrases)。最简单的情况是用关键词来描述聚类文档,更多的方法则采用了关键词短语来进行类目描述,因为关键词短语比词表达的信息更加丰富。根据聚类算法可将聚类浏览技术分为扁平聚类(Flat Clustering)和层次聚类(Hierarchical Clustering)。扁平聚类只对数据进行一层的划分,类目的组成比较粗略,层次聚类自动将产生的类目组织成树形结构以便于用户浏览。

4　聚类浏览技术的主要算法

聚类和标识是 Web 聚类浏览系统的两个基本组成部分,但目前提出的方法各有侧重。一些方法将标识提取作为主要目标,在标识提取的过程中形成聚类。另外一些方法则将对信息的聚类作为最重要的步骤,标识短语严格依靠产生的类目[11]。以下将讨论聚类浏览技术中常用的聚类算法及改进方法。

4.1　传统聚类算法的应用

文献中有关文本聚类的算法很多。层次聚合算

法AHC(Agglomerative Hierarchical Clustering)能够生成类目层次体系,但对于文档集合数量很大时,处理速度很慢,最短距离法(Single-link)和类平均法(Group-average)的时间复杂度为$O(n^2)$,n为文档个数,最长距离法(Complete-link)的时间复杂度为$O(n^3)$。而且AHC算法对聚类停止标准非常敏感,容易产生对用户来说没有意义的分类。

线性时间聚类算法满足在线聚类的速度要求,典型算法是K-均值聚类算法(K-Means),其时间复杂度为$O(nkT)$,k是类别个数,T为迭代次数。与AHC不同的是,K-Means算法能产生重叠聚类,但该算法的执行结果与文档输入顺序有关。WEBCAT[13]系统采用了K-Means算法产生扁平聚类。

Scatter/Gather技术采用了Buckshot和Fractionation两种聚类算法进行快速聚类,其时间复杂度为$O(kn)$。两者相比,Fractionation生成类的准确性要高,而Buckshot的速度则更快,更适宜于实时的聚类。SCATTER/GATHER[14,15]是最早应用在搜索引擎之上的Web聚类软件之一,它采用Scatter/Gather技术来组织和浏览文档。

这些聚类浏览系统大多采用向量空间模型,单纯利用词频信息构造相似度矩阵,按文档的相似度进行聚类,并抽取成员文档的公共词项作为聚类的标识。除了传统聚类算法本身的不足外,聚类标识的可读性较差,使得用户难以识别相关的文档类别。

4.2 后缀树聚类算法STC(Suffix Tree Clustering)

后缀树算法是由Zamir等人提出的增量式聚类算法[10,16],把文档看作有顺序的词串(Phrase)而不仅仅是一个单词的集合,并通过文档共现的词串来作为文档相似度测量的基础。如两个文档共有至少一个的词串,则将它们合并为一个基类。STC在表示文档间相似度时,采用的是后缀树的结构[17]。

(1) 确定基类

STC方法对文档集合中的句子进行标引,形成后缀树的结构。每个句子看成是由n个有序单词组成。句子用SE_i表示,其包含的单词为W_{i1},W_{i2},…,W_{in},则子串W_{i1},…,W_{in}($1 \leq j \leq n$)是SE_i的后缀。可以看出有n个单词的句子有n个后缀,它们分别从第1个单词、第2个单词开始。在构筑的后缀树中,每一个叶节点是由从根到此节点的后缀所标识的,并用这些后缀所在的文档对叶节点进行标注。同样,从根到非终端节点的路径代表的是前缀,一个前缀可以有两个或更多的后缀。非终端节点是由它的前缀所标识的,用它所含有的后缀所在的文档对其进行标注。后缀树中含有至少两个文档的节点即为基类,节点的标识代表文档的公共词串,可作为聚类标识。

由STC生成的基类,可以通过以下的求值函数来求出每一个基类的值。

$$S(B) = |B| \times f(|B|) \times \sum tfidf(w_i) \quad (1)$$

其中,$S(B)$表示基类B的值,$|B|$表示词串的单词数目,即词串长度,$f(|B|)$表示词串规范化处理长度,$tfidf(w_i)$表示词频调整值。

(2) 合并相似基类

相似基类是指它们所含的文档集合有很多的重叠,为了避免出现几乎相同的类目,需要对相似基类进行合并。

两个基类B_m和B_n,$|B_m|$和$|B_n|$表示基类的长度,即基类所包含的文档数目。$|B_m \cap B_n|$表示基类B_m和基类B_n所共有的文档数目。

$$|B_m \cap B_n| / |B_m| > \alpha \quad (2)$$

$$|B_m \cap B_n| / |B_n| > \alpha \quad (3)$$

当满足公式(2)和(3)时,说明B_m和B_n相似,可以归为一类。式中的α为基类合并阈值。Zamir等人提出的阈值为0.5。

当检索结果数量巨大时,可以按基类的值进行排序,只对排在前k位的基类进行相似度计算和相应的合并处理。

后缀树算法是线性时间聚类算法,时间复杂度为$O(n)$。其聚类过程随文档的依次输入实时进行处理,且聚类的结果同文档的处理顺序无关。它是一个基于短语的方法,把文档看作有顺序的单词串而不仅仅是一个单词的集合,因为充分利用了单词之间的信息,从而提高了聚类的质量。

Dell Zhang[18]提出了一种基于语义的层次在线信息检索聚类方法SHOC,对Zomir等人的工作进行了扩展。由于后缀树结构的性能(时间复杂度和空间复杂度)与文本语言的字母表大小相关[19],不适用于对中文文档的处理。为此,提出了基于后缀树组的关键词短语提取算法,在空间复杂度上得到极大的改善,并能够避免提取无意义的部分短语,同时支持中文和英文两种语言。后缀树算法只是简单将共享相同短语的文档归为一类,而没有考虑到自然语言中存在的同义词和多义词现象,从而使得聚类不够准确和完整。针对这一问题,基于潜在语义标引(Latent Semantic Indexing,LSI)[20]的思想,利用奇异值分解SVD方法来捕捉词项—文档矩阵中

的"潜在"结构,实现正交聚类,将搜索结果聚类形成基于语义的树形结构,便于用户浏览。张健沛等提出了一个基于修改的 PAT-tree 数据结构的中文搜索引擎结果聚类算法[21],该算法把 PAT-tree 数据结构和 STC 结合起来用于中文文档聚类,它使用 PAT-tree 数据结构克服 Suffix tree 处理中文信息的不足,使用 STC 框架来保证聚类能有效地执行。Irmina Maslowska 则在 STC 的基础之上提出了层次后缀树聚类算法[22],采用有向图来表示基类之间的关系,对有向图进行裁剪建立层次分类。

4.3　Carrot2 系统

Grouper[9] 系统是第一个实现 web-snippet 聚类的软件,采用了后缀树聚类算法。Vivisimo[23] 是一个商业的搜索结果聚类引擎,它采用了一种特定的启发式算法,是目前效果较好的聚类搜索引擎。针对这两种系统只能应用于英文检索,并且受到 Grouper 工作的启示,Weiss[24] 实现了后缀树算法的开源聚类系统——Carrot2,并增加了对波兰语言的支持。Carrot2 采用了开源、模块化的结构,可以作为其他研究工作者研究的基础,该系统作为一个研究框架,为其他聚类算法的研究提供了实验的测试床,能够实现对各种数据源的自动查询,处理搜索结果及聚类结果的可视化输出。因此,Carrot2 的发展对搜索结果聚类的研究产生了极大的推动力,并由此产生了一些新的算法,如 LINGO[25],AHC[26]。

LINGO 算法受到 SHOC 的启发,并基于与 Vivisimo 系统相似的思想,即"先形成可描述的类"。该算法采用 SVD 方法从词项—文档空间中形成概念—文档空间,利用向量空间模型中的文档相似度计算方法,抽取与概念最近的词项作为该概念的标识,每个概念及其标识形成一个类,最后将文档分配到与概念空间中的聚类标识最近的类中。由于 LSI 能够发现词项—文档之间的内在关系,这种方法产生的聚类标识能够较好地表达类别的内容。LINGO 算法作为一个组件集成在了 Carrot2 系统框架中,目前是 Carrot2 演示版的默认聚类算法。但是 LINGO 算法的速度取决于 SVD 计算的复杂度,因而在应用于大量的搜索结果时比较耗时。

4.4　SnakeT 系统

SnakeT 是 Ferragina 和 Gulli[27,28] 提出的具有层次结构和短语标识的聚类系统,通过类似频繁项集的方法抽取有意义的标识,采用自底向上的层次聚类算法对这些频繁短语进行聚类。

(1) 短语标识的选择和评级

聚类标识是从 snippets 中抽取的非相邻词项序列,称为 gapped sentence,这一方法克服了 Grouper 系统中对标识的相邻性限制。为了提高选择标识的质量,利用了两个数据库。前者提供了链接和链接描述文字信息,用于丰富 snippets 的内容,提供更好的候选短语。后者由 DMOZ 的开放式目录(Open Directory Project)组成,根据候选短语在 DMOZ 类目中出现的位置和频率采用 IF×IDF 方法对其进行评级。

$$TF(w) = 1 + \log \#(w) \tag{4}$$

$$IDF(w) = \log \frac{\#c}{\#c(w)} \tag{5}$$

其中,$\#(w)$ 为词项 w 在 DMOZ 中出现的总次数,$\#c(w)$ 为 w 出现的 DMOZ 类目数,$\#c$ 是 DMOZ 类目总数。

词项 w 关于类 C_i 的评分:

$$rank(w, C_i) = b(w, C_i) \times TF(w) \times IDF(w) \times ns(C_i) \tag{6}$$

其中,$ns(C_i)$ 表示类目 C_i 的增强因子,$b(w, C_i)$ 表示词项 w 出现在 C_i 相关位置中的增强因子。

词对 (w_h, w_k) 的评分:

$$rank(w_h, w_k) = \max_{C_i}\left\{\prod_{r=h,k} b(w_r, C_i) \times TF(w_r) \times IDF(w_r) \times ns(C_i)\right\} \tag{7}$$

从 snippets 中抽取频繁出现的词对(可看作频繁短语),根据式(7)进行评分,将评分较高的词对合并构成 gapped sentence。丢弃评分较低的 gapped sentence 后形成层次聚类叶节点的候选标识。

(2) 层次聚类

SnakeT 采用自底向上的层次聚类算法。snippets 按照共享的候选标识组成初始类,作为层次聚类的叶节点,它们的标识则称之为初始标识。选择在初始类 C 的 snippets 中出现频率至少为 c% (SnakeT 中 $c = 80$) 的候选标识作为类 C 的第二标识对其进行扩展,初始标识是对类 C 的详细描述,第二标识则是对类 C 的粗略描述,它们共同构成类 C 的完全描述。依此思路从初始类向上逐步构建父类信息和标识。最后对形成的层次聚类进行必要的剪枝和合并。

SnakeT 系统是第一个具有层次结构和短语标识的开源 snippets 聚类系统,其性能评价接近

Vivisimo 系统。

4.5 基于继承机制的 CHCA 算法

CHCA 算法[29] (Cluster Hierarchy Construction Algorithm) 是基于词项—文档的二进制矩阵(称为成员表),采用面向对象中的继承机制建立层次聚类。每一个文档都可能成为一个类,它包含的词项是其属性。在自然语言中,当用更多的词来描述一个实体时,该实体就更特殊。因此一个类具有更多关联词项时就更具体化,某个类若包含该类的部分属性,则成为它的父类。聚类过程从最普通的类(含有最少词项的文档)开始,采用迭代的方法将每篇文档分配到一个类中,最后将类进行合并,每个类由相关的词项集加以描述。文档分配的过程中需要计算文档与类之间的距离。

$$Dis(r,i) = \frac{1}{m}\sum_{j}^{m} Y_{ij} \times (Y_{ij} - X_{rj}) \qquad (8)$$

其中,m 是成员表中的词项(列)数,X_{rj} 是成员表中文档 r 的第 j 列元素,Y_{ij} 是类 i 向量中的第 j 个元素。

根据公式(8)将文档 r 分配给具有最小距离的类 i。CHCA 是一种快速算法,并且支持多继承,可实现重叠聚类。

4.6 短语评分法

这类方法首先提取文档短语,对每个短语计算相关属性特征值,进行合并赋予一个评分,根据评分抽取关键概念形成聚类。WISE[30] 中利用决策树模型对各短语进行评价,具有最相关评分的短语构成关键概念向量来表示文档。最后采用基于图论的聚类算法 PoBOC[31] 进行聚类,将含有不同含义的关键概念聚集到不同的类,以实现重叠聚类,并选择发生频率最高的关键概念作为聚类标识。文献[32]则将聚类问题转化为显著短语排队问题,利用回归模型得到每个短语的显著评分。按照评分将短语排序,排在前面的短语成为显著短语,含有显著短语的文档则构成候选类,显著短语作为聚类的标识。WISE 系统是对整个网页进行语义分析,因而比较耗时,同时 PoBOC 算法并不适用于大的文档集合。文献[32]是对搜索结果文档摘要进行短语抽取,但形成的是扁平聚类,可浏览性不是很好。

4.7 基于概念格的聚类算法

形式概念分析(Formal Concept Analysis, FCA)理论用于概念的发现、排序和显示,概念格是形式概念分析中核心的数据结构,表明了概念间泛化和例化之间的关系,概念格建格的过程就是概念聚类的过程。根据概念格的特点,已提出一些将概念格应用于搜索结果聚类浏览的算法。CREDO[33] 系统是利用概念格方法的在线聚类系统,采用关键词标识和层次结构。在概念格的建格过程中,由于文档词项太多会导致许多不相关的概念,而过分限制文档词项又会使许多文档缺乏共享词项而不能归类,为此,在 CREDO 系统中,概念格的建立是两层的分类过程,概念格顶层元素只通过搜索结果标题中的词项来构建以区分其主要主题,较低层的节点则增加了文档片断中的词项来扩展其外延,包含了每个主题下的子主题。文献[34]为了处理聚类过程中的模糊信息,提出了基于模糊概念格的概念聚类方法,解决了概念聚类中概念间的多重继承关系,并应用到 Web 搜索结果聚类上。

4.8 其他方法

文献[35～37]将数据挖掘中的频繁项集概念应用于文档聚类,利用关联规则挖掘算法找出文档集的频繁项集,以此作为文档聚类的依据。采用关联规则算法最大的问题在于需要下载整个网页进行分析,因而耗时较长,效率不高,无法满足 snippets 聚类的要求。Kammamuru 等人提出了一种基于单一特征的层次聚类算法[38],目的是为了最大化单一特征所描述的文档覆盖率和聚类标题的区分能力。文献[39]把聚类看作对搜索结果建立结构化标识列表的过程。利用命名实体(Named Entity, NE)提取工具从文档中抽取词项作为候选标识,并根据 NE 抽取过程中确定的分类信息对标识进行分类。这种方法的应用范围直接受 NE 分类结构的影响。Lawrie 和 Croft[40] 则把聚类/标识问题作为文档集的多级摘要产生过程,认为最好的主题摘要词项既与主题相关又可预测其他词项。该技术基于离线建立的统计语言模型,通过计算每个词项的交叉熵(Kullback-Leibler Divergence)来估计其主题性,并表明此方法优于通过文档集的 $tf \times idf$ 来选择词项。

5 聚类评价方法

对于搜索引擎聚类浏览技术,由于缺乏标准评价数据集和性能衡量标准,评价一直是一个难题,尤其是对聚类标识的评价。

文献[41]采用了文档聚类中的 F 值评分作为搜索结果聚类的评价标准,该方法需要采用聚类基准,而对于搜索引擎的返回文档集而言该基准往往是未知的。

针对搜索结果聚类的特点,人们提出了一些新的评价方法。

Wang[42]提出平均信息熵的评价方法。信息熵用来衡量聚类的纯度,旨在判定同类中的网页是否真正是关于同一个主题的。

聚类后形成的任一类别 j 的信息熵定义为:

$$E(j) = -\sum p_{ij}\log(p_{ij}) \qquad (9)$$

其中, p_{ij} 是类别 j 属于给定类 i 的概率。

聚类结果集的平均信息熵定义为:

$$E = \sum_{j}^{m}\frac{n_j \times E(j)}{n} \qquad (10)$$

其中, n_j 是类别 j 的大小, m 是聚类的类别总数, n 是聚类的网页总数。

Zeng[32]将聚类问题转化为显著短语排队问题,并提出采用类似信息检索中的 $P@N$ 方法对聚类性能进行评价。

$$P@N = \frac{|C \cap R|}{|R|} \qquad (11)$$

其中, R 是前 N 个显著短语集合, C 是人工标识正确的显著短语集合。

这种评价方法具有很强的创新性和实用性, SnakeT 系统中对该方法进行了扩展,用于评价层次聚类的标识。

用户评价方法包括系统日志分析和用户主观评价方式。Grouper 通过对系统长期的日志进行分析,根据统计结果对聚类性能做出评价。LINGO 则采用了用户主观评价法,即通过问卷调查的方式,根据对测试用户的反馈进行分析来完成对聚类系统性能的评价。这种方法也是目前采用较多的一种评价方法。

6　结束语

聚类浏览技术增强了搜索引擎的可浏览性,利于用户快速定位到感兴趣的信息,已引起越来越多的研究人员的关注。由于传统的聚类算法在应用中受到一些限制,因而不断有新的聚类算法提出。目前的算法中还存在一些问题,需要进一步进行研究。

(1) 算法的效率问题: 由于搜索引擎中在线聚类的特点,使得提高聚类算法的效率问题显得至关重要。不仅要有增量聚类的能力,有较好的伸缩性,而且要能适用于大的文档集合。

(2) 聚类的标识问题: 语义清晰的标题对用户的浏览具有引导作用,好的聚类标识能使用户快速了解网页的主题内容。由于聚类是基于搜索引擎返回的结果,如何从简短的信息中提取词项准确描述网页内容仍是一个巨大的挑战。同时,在实际应用中,如何平衡聚类速度和聚类标识的准确性也是值得探讨的问题。

(3) 层次结构聚类算法的研究: 搜索结果的层次结构更利于用户浏览聚类信息,如何建立合理的层次结构及自动发现适当的类别粒度仍然是目前的研究热点,结合 Web 分类目录和领域本体知识成为人们关注的一个方向。

参考文献:

[1] Pretschner A, Gauch S. Ontology Based Personalized Search[A]. In: Proceedings of the Eleventh IEEE International Conference on Tools with Artificial Intelligence[C]. 1999. 391-398.

[2] Jansen B J, Spink A, Bateman J, Saracevic T. Real Life Information Retrieval: A Study of User Queries on the Web[J]. ACM SIGIR Forum, 1998, 32(1): 5-17.

[3] Franzen K, Karlgren J. Verbosity and Interface Design[A]. Technical Report T2000:04, Swedish Institute of Computer Science(SICS)[C]. 2000.

[4] Chen H, Dumais S. Bringing Order to the Web:Automatically Categorizing Search Results[A]. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems[C]. New York: ACM Press, 2000. 145-152.

[5] Kules B, Kustanowitz J, Shneiderman B. Categorizing Web Search Results into Meaningful and Stable Categories Using Fast-Feature Techniques[A]. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries[C]. New York: ACM Press, 2006. 210-219.

[6] Cui H, Zalane O R. Hierarchical Structural Approach to Improving the Browsability of Web Search Engine Results[J]. IEEE, 2001, 956-960.

[7] Griffiths A, Luchhurst H, Willett P. Using Inter-Document Similarity Information in Document Retrieval Systems[J]. Journal of the American Society for Information Sciences, 1986, 37:3-11.

[8] Salton G. The SMART Retrieval Systems[M]. Prentice Hall, Englewood Cliffs, N.J., 1971.

[9] Zamir O, Etzioni O. Grouper: A Dynamic Clustering Iinterface to Web Search Results[A]. In: Proceedings of the Eighth International World Wide Web Conference(WWW8)[C]. 1999.

[10] Zamir O, Etzioni O. Web Document Clustering: A Feasibility Demonstration[A]. In: Proceeding of the 19th International ACM SIGIR Conference on Research and Development of Information Retrieval (SIGIR'98)[C]. New York: ACM Press, 1998, 46-54.

[11] Geraci F, Pellegrini M, Maggini M, Sebastiani F. Cluster Generation and Cluster Labeling for Web Snippets[A]. In: SPIRE 2006[C]. Berlin / Heidelberg: Springer, 2006(4209):25-36.

[12] 刘远超, 王晓龙, 徐志明, 等. 文档聚类综述. 中文信息学报, 2006, 20(3):55-62.

[13] Giannotti F, Nanni M, Pedreschi D. Webcat: Automatic Categorization of Web Search Results[A]. In: SEBD03[C]. New York: ACM Press, 2003. 507-518.

[14] Cutting D R, Karger D R, Pedersen J O. Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collection[A]. In: Proceedings of the 16th Annual International ACM/SIGIR Conference on Research and Development in Information Retireval[C]. New York: ACM Press, 1993. 125-135.

[15] Hearst M A, Pedersen J O. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results[A]. In: Proceedings of the 19th Annual International ACM/SIGIR Conference on Research and Development in Information Retireval[C]. New York: ACM Press, 1996. 76-84.

[16] Zamir O, Etzioni O, Madani O, et al. Fast and Intuitive Clustering of Web Documents[A]. In: Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining[C]. 1997. 287-290.

[17] 孙建军, 成颖, 等. 信息检索技术[M]. 北京:科学出版社, 2004, 223-228.

[18] Zhang D, Dong Y. Semantic, Hierarchical, Online Clustering of Web Search Results[A]. In: Proceeding of the 6th Asia Pacific Web Conference(APWEB)[C]. 2004. 69-78.

[19] Manber U, Myers G. Suffix Arrays: A New Method for On-line String Searches[A]. In: Proceedings of the first Annual ACM-SIAM Symposium on Discrete Algorityms[C]. 1990. 319-327.

[20] Deerwester S, Dumais S, Furnas G, et al. Indexing by Latent Semantic Analysis[J]. Journal of the American Society for Information Science, 1990, 41: 391-407.

[21] 张健沛, 刘洋, 杨静, 等. 搜索引擎结果聚类研究[J]. 计算机工程, 2004, 30(5):95-97.

[22] Maslowska I. Phrase-Based Hierarchical Clustering of Web Search Results[A]. In: Proceedings of the 25th European Conference on IR Research[C]. Pisa, Italy: Springer, 2003. 555-562.

[23] Vivisimo. http://vivisimo.com/faq/technology.html.

[24] Weiss, D. A Clustering Interface for Web Search Results in Polish and English[D]. Poznan University of Technology, Poland, June 2001.

[25] Osinski S. An Algorithm for Clustering of Web Search Results[D]. Poznan University of Technology, Poland, June 2003.

[26] Wroblewski, M. A Hierarchical WWW Pages Clustering Algorithm Based on the Vector Space Model[D]. Poznan University of Technology, Poland, July 2003.

[27] Ferragina P, Gulli A. A Personalized Search Engine Based on Web-Snippet Hierarchical Clustering[A]. In: Special Interest Tracks and Poster Proceedings 14th International Conference on the World Wide Web[C]. New York: ACM Press, 2005. 801-810.

[28] Ferragina P, Gulli A. The Anatomy of SankeT: A Hierarchical Clustering Engine for Web-Page Snippets[A]. In: PKDD 2004[C]. Berlin / Heidelberg: Springer, 2004. 506-508.

[29] Schenker A, Last M, Kandel A. Design and Implementation of a Web Mining System for Organizing Search Engine Results[J]. International Journal of Intelligent Systems, 2005. 20: 607-625.

[30] Campos R, Dias G, Nunes C. WISE: Hierarchical Soft Clustering of Web Page Search Results Based on Web Content Mining Techniques[A]. In: Proceeding of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence[C]. Washington, DC, USA: IEEE Computer Society, 2006. 301-304.

[31] Cleuziou G, Martin L, Vrain C. PoBOC: an Overlapping Clustering Algorithm, Application to Rule-Based Classification and Textual Data[A]. In: Proceedings of the 16th European Conference on Artificial Intelligence ECAI[C]. 2004. 22-27.

[32] Zeng HJ, He QC, Chen Z, et al. Learning to Cluster Web Search Results[A]. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval[C]. New York: ACM Press, 2004. 210-217.

[33] Carpineto C, Romano G. Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO[J]. Journal of the Universal Computer Science, 2004, 10(8): 985-1013.

[34] 黄建斌, 姬红兵. 基于模糊概念格的 Web 搜索结果聚类算法[J]. 西安电子科技大学学报, 2005, 32

(6) :856-860.

[35] Fung B C M, Wang K, Ester M. Hierarchical Document Clustering Using Frequent Itemsets [A]. In: Proceedings of the 2003 SIAM International Conference on Data Mining[C]. 2003.

[36] Li F, Mehlitz M, Feng L, et al. Web Page Clustering and Concept Mining: An Approach towards Intelligent Information Retrieval [J]. Cybernetics and Intelligent Systems 2006, 1-6.

[37] 宋春芳, 石冰. 一种基于关联规则的搜索引擎结果聚类算法[J]. 山东大学学报, 2006, 41(3): 61-65.

[38] Kummamuru K, Lotlikar R, Roy S, et al. A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results[A]. In: Proceeding of 13th International Conference on World Wide Web[C]. New York: ACM Press, 2004. 658-665.

[39] Toda H, Kataoka R. A Search Result Clustering Method using Informatively Named Entities[A]. In: Proceedings of the 7th annual ACM international workshop on Web information and data management [C]. New York: ACM Press, 2005. 81-86.

[40] Lawrie D J, Croft W B. Generating Hierarchical Summaries for Web Searches[A]. In: Proceedings of SIGIR-03, 26th ACM International Conference on Research and Development in Information Retrieval[C]. New York: ACM Press, 2003. 457-458.

[41] Chuang SL, Chien L F. A Practical Web-Based Approach to Generating Topic Hierarchy for Text Segments[A]. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management [C]. New York: ACM Press, 2004. 127-136.

[42] Wang Y, Kitsuregawa M. Evaluating Contents-Link Coupled Web Page Clustering for Web Search Results [A]. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management[C]. New York: ACM Press, 2002. 499-506.

(上接第 49 页)

[5] ZHAO S B, GRISMAN R. Extracting relations with integrated information using kernel methods [A]. ACL'2005[C]. USA: 25-30 UnivofMichgan-AnnArbor June 2005. 419-426.

[6] ACE 2004. The Automatic Content Extraction (ACE) Projects, 2007 (2007-4-20). http//www.ldc.upenn.edu/Projects/ACE/.

[7] WANG T, LI Y Y, KALINA B, et al. Automatic Extraction of Hierarchical Relations from Text[A]. Proceedings of the Third European Semantic Web Conference (ESWC 2006) [C]. USA: Springer, 2006. 401-416.

[8] ZHANG M, ZHANG J, SU J, et al. A Composite Kernel to Extract Relations between Entities with both Flat and Structured Features [A]. ACL'2006 [C]. Sydney: July, 2006. 825-832.

[9] 车万翔, 刘挺, 李生. 实体关系自动抽取[J]. 中文信息学报, 2005, 19(2): 1-6.

[10] 董静, 孙乐, 冯元勇, 黄瑞红. 中文实体关系抽取中的特征选择研究[J]. 中文信息学报, 2007, 21(4): 80-85.

[11] Charniak E. A maximum-entropy-inspired parser[A]. Nirenburg S, ed. Proc. of the NAACL 2000 [C]. Washington: ACL, 2000. 132-139.