基于词项图分析的查询分面挖掘方法

窦志成[1] 江政宝[1] 李谨秀[1] 张宜春[3] 文继荣[1),2)]

[1)](中国人民大学信息学院 北京 100872)

[2)](中国人民大学大数据管理与分析方法研究北京市重点实验室 北京 100872)

[3)](中国艺术科技研究所 北京 100012)

摘 要 查询分面是用于描述查询某一方面内容的一组并列的词或词组.现有的查询分面挖掘方法主要通过模式挖掘搜索结果中包含的高频列表,并利用无监督或有监督的方法对高频列表进行聚类,最终得到查询分面.因为通常采用的搜索结果的数目有限,这种方法挖掘出的查询分面及其包含的分面项的覆盖率不高.针对这一问题,该文提出了一种基于从大规模网页中构建的词项图的查询分面挖掘方法.首先基于大规模网页数据集构建词项图,图中的节点代表词项,边代表两个词项的相似性.针对每个查询,从搜索结果中挖掘出初始分面,然后基于词项图对这些初始查询分面进行扩充,找到词项图中与初始分面类似的候选词,对候选词抽取多种特征,最后利用支持向量机对候选词进行分类,预测词项是否可为扩充词项,并将预测为正例的词项扩充到分面中.该扩充过程迭代多次直到无法找到更多分面项.实验表明该方法可有效提高查询分面的质量,尤其是能够显著改善分面项的覆盖率.

关键词 查询分面;用户意图;频繁列表;词项图;知识库;社会媒体;社会计算

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2017.00556

A Method of Mining Query Facets Based on Term Graph Analysis

DOU Zhi-Cheng[1] JIANG Zheng-Bao[1] LI Jin-Xiu[1] ZHANG Yi-Chun[3] WEN Ji-Rong[1),2)]

[1)](*School of Information, Renmin University of China, Beijing* 100872)

[2)](*Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing* 100872)

[3)](*China Art Science and Technology Institute, Beijing* 100012)

Abstract A query facet is a list of homogeneous words or phrases that can describe an underlying aspect of the query. Existing algorithms use predefined patterns to extract frequent lists contained in the top search results of the query, then group these lists into clusters by using unsupervised or supervised learning methods to generate final query facets. The coverage of query facets and their items mined by these methods might be limited, because only a small number of search results are used. In order to solve this problem, we propose mining query facets by using a term graph constructed from a large number of web pages. The nodes in this graph represent different terms and the edges represent the similarity between terms. We first mine initial query facets from the top search results of the query, then find similar terms from the term graph as candidates. Different features of each candidate are extracted. Finally we use support vector machine to classify all candidates into two sets, namely positive set and negative set. All the positive terms are used to expand initial query facets. These steps are repeated until no more facet items are found. Experimental

收稿日期:2015-09-15;在线出版日期:2016-04-24.本课题得到国家自然科学基金(61502501)和国家"九七三"重点基础研究发展规划项目基金(2014CB340403)资助.窦志成,男,1980年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为信息检索、数据挖掘、大数据等.E-mail:dou@ruc.edu.cn.江政宝,男,1993年生,硕士研究生,主要研究方向为数据挖掘、信息检索等.李瑾秀,女,1991年生,硕士研究生,主要研究方向为数据挖掘.张宜春,男,1978年生,博士,主要研究方向为艺术表现相关的数据与信息处理、文化科技相关应用的基础理论研究.文继荣,男,1972年生,博士,教授,国家千人计划专家,中国计算机学会(CCF)会员,主要研究领域为信息检索、数据库、大数据、数据挖掘.

results show that the proposed method can significantly improve the quality of mined query facets, and it can especially improve the coverage of facet items.

Keywords　query facets; user intent; frequent lists; term graph; knowledge base; social media; social computing

1　引　言

据中国互联网络信息中心(CNNIC)发布的《第35次中国互联网络发展状况统计报告》[1]显示，截至2014年12月，我国搜索引擎用户规模达52 223万人，网民使用率为80.5%. 手机搜索引擎用户规模达42 914万人，手机网民使用率达到77.1%. 搜索引擎是中国网民除即时通信外使用率最高的互联网应用. 这表明搜索引擎是人们从互联网获取信息的一个必不可少的工具，在人们日常生活中发挥了重要作用.

虽然已经被广泛使用，但现有的搜索引擎仍然采用以返回简单结果列表为主的检索方式. 随着互联网的高速发展，互联网上信息的复杂度和多样性也越来越高，用户的信息获取需求也越来越复杂多样. 在很多情况下，简单的搜索结果列表已经无法满足用户的高阶信息需求. 例如，用户想要购买一块手表，而在购买之前想用搜索引擎来全方面了解手表的品牌、特征、性能、型号等各方面信息. 在目前的搜索方式下，用户需要点击并浏览大量网页才能总结出上述知识. 如果能够自动帮助用户汇总生成上述信息并显示给用户，则可能会大大减少用户要花费的时间，有效提高查询效率和用户满意度. 同时，当用户任务复杂、查询需求不清晰或者不确定时，往往需要多次修改查询词才能最终确定想要查找的内容. 目前的查询推荐虽然能够在一定程度上满足用户修改查询词的需求，但因为推荐的查询结构化不强，各种内容混杂在一起，不方便用户查阅，迫切需要更好的方式来支持用户的探索式搜索需求.

解决上述问题的一个有效途径是进行查询分面(Query Facets)挖掘. 和购物搜索引擎中的分面类似，一个查询分面由一组语义相关的并列词项(Facet Item)组成. 一个查询分面整体上概括和描述了查询所涵盖的某一方面或维度的内容. 一个查询可具有多个分面，分别刻画查询在不同方面的信息. 表1中给出了从文本中自动挖掘的一些查询分面. 对于查询“手机”，第1个分面为手机品牌，包含三星、苹果、HTC等. 第2个分面为手机类型，如直板手机、翻盖手机、商务手机等. 第3个分面为网络类型，第4个分面包含了手机相关的一些主题. 而对于查询“中国人民大学”，则可自动挖掘出关于中国人民大学的学院、学科、校长以及相关院校等内容的分面. 对于“北京的写字楼”，则自动挖掘出地区、商圈、出租出售类型以及写字楼名字等. 查询分面是查询内容的总结和概括，一方面可以直接满足用户获取高阶知识的需要，例如用户可直接获取到手机的重要品牌，而不需要自己再进行大量的浏览和总结工作. 另一方面，基于查询分面可实现互联网分面搜索(Faceted Search). 和在购物搜索中被广泛采用的分面搜索类似，为互联网搜索提供分面搜索功能可帮助用户快速地定位到其所需要的信息，更好地帮助用户进行探索式搜索，有效提高用户体验. 例如，图1显示的为查询“中国人民大学”的查询分面挖掘结果以及基于这一结果实现的简单分面搜索的示意图. 如前所述，对于该查询，可自动挖掘出关于中国

表1　查询分面示例

查询	序号	分面s
手机	1	三星，苹果，htc，诺基亚，华为，联想，小米，索尼，摩托罗拉，酷派，魅族，中兴，oppo，lg，天语，…
	2	直板手机，翻盖手机，商务手机，音乐手机，拍照手机，三防手机，学生手机，全键盘手机，时尚手机，老人手机，儿童手机，…
	3	移动3g，联通3g，电信3g，移动4g，gsm，4g，cdma，双模，移动联通2g，电信2g，…
	4	手机游戏，手机主题，手机软件，手机铃声，手机图片，手机壁纸，手机网游，…
北京的写字楼	1	朝阳，海淀，东城，通州，大兴，西城，昌平，崇文，丰台，宣武，…
	2	金融街，中关村，东二环，cbd，东三环，国贸，望京，东城区，国贸cbd，泛cbd，上地，…
	3	出租，出售，租售，转让
	4	金地中心，中环世贸中心，北京财富中心，嘉铭中心，华贸中心，中海广场，…
中国人民大学	1	经济学院，财政金融学院，法学院，商学院，哲学院，统计学院，新闻学院，环境学院，信息学院，公共管理学院，…
	2	北京大学，清华大学，北京师范大学，北京航空航天大学，北京理工大学，北京交通大学，对外经济贸易大学，中国政法大学，…
	3	理论经济学，应用经济学，法学，政治学，社会学，新闻传播学，哲学，金融学，管理学，心理学，…
	4	成仿吾，袁宝华，黄达，吴玉章，李文海，郭影秋，纪宝成，陈雨露，…

图 1 查询"中国人民大学"的分面搜索界面示意图(界面截图取自 2015 年 10 月)

人民大学的学院、学科、校长以及相关院校等内容的分面并显示给用户. 用户可以进一步通过选择一个或多个分面项(如信息学院),对检索结果进行过滤(如查找关于信息学院的信息),进而满足快速查找某一子查询相关信息的需求.

值得一提的是,和购物搜索或者其他搜索中的预定义分面类型(例如购物搜索引擎中广泛使用品牌、颜色、价格范围等)不同的是,本文中研究的分面挖掘方法应用在 Web 搜索引擎中,不基于或利用任何领域知识和数据库,不需要人工编辑. 从上面举出的"手机"、"中国人民大学"、"中关村附近的写字楼"的分面结果可以看出,不同查询下查询分面的类型差异很大,没有预定义模式可用,这也是查询分面挖掘面临的难点之一.

现有的查询分面挖掘方法 QDMiner[2] 和 QF-I、QF-J[3] 主要是通过挖掘查询的搜索结果集中包含的频繁列表来进行. 这些方法巧妙地利用了网站设计时并列信息(如品牌列表)经常以列表(如下拉框、表格、并列词组等)的形式在网站上展示的这一特点,通过对查询的检索结果中的列表进行抽取、聚类和排序来生成查询分面. 一般来说,查询相关的重要信息列表会被多个网站频繁使用. 基于这一思想,现有查询分面挖掘方法主要通过查询分面中包含的列表在搜索结果中出现的频度来对查询分面的重要度进行估计. 虽然实验表明上述方法可以有效地挖掘出高质量的查询分面[2-3],但因为主要是基于搜索结果中的信息列表来生成分面,分面的准确性和覆盖率很大程度上受限于查询结果集以及其中包含的列表的数量和质量. 若查询的某类信息不以列表的形式或其他任何形式在搜索结果中展现,则上述算法就无法挖掘出对应的分面信息或者导致挖掘到的分面中词项残缺不全.

针对这一问题,本文提出一种基于从大规模互联网语料中挖掘的词项图(Term Graph)来挖掘和完善查询分面的方法 $\mathrm{QDMiner}_{\mathrm{TG}}$. 词项图中的节点为互联网上出现的词或者词组,而两个节点之间的边表示这两个节点之间的语义对等或者相似程度. 利用大规模词项图一方面可以有效解决词项不全的问题,同时也可以进一步利用大规模互联网网页中词项的共现关系来剔除分面中的噪声词项,提高分面质量. 本文提出的算法 $\mathrm{QDMiner}_{\mathrm{TG}}$ 的基本思路是:首先从检索结果集中挖掘出初始查询分面,然后采用大规模词项图对初始查询分面进行扩充,为分面中的词项查找到类似词项,然后将这些新发现的词项与初始查询分面进行融合进而生成新的分面. 然后再对完善后的查询分面重新进行扩充. 通过多次迭代的方法,逐步提高查询分面的质量. 实验表明该方法可有效提高查询分面的质量,和现有的从搜索结果集中挖掘查询分面的方法相比,不但可以挖掘出更多的词项,显著提高查询分面中词项的召回

率；同时可以发现原分面中存在的低质量词项，提高分面精确度和排序质量.

本文第 2 节将介绍相关研究工作，包括现有查询分面挖掘算法、集合扩展算法以及查询推荐算法；第 3 节详细介绍本文提出的查询分面挖掘方法；第 4 节将进行实验并对实验结果进行分析；最后对全文内容进行总结并对未来工作进行展望.

2　相关工作

2.1　查询分面挖掘

窦志成等人[2]首次明确提出了查询分面挖掘的问题，并提出 QDMiner 方法. 该方法通过挖掘查询搜索结果中包含的频繁列表，自动生成查询分面. 该方法假设查询词所覆盖的重要信息一般会在网页中以列表的形式进行展现，因此通过对相似列表进行加权和聚类，即可挖掘出和查询相关的各组分面. 该方法不利用任何预定义领域知识，适用于任何类别和领域的查询. 孔维泽等人[3-4]在此基础上提出了基于监督学习生成查询分面的方法，即 QF-I 和 QF-J. 该方法使用和 QDMiner 类似的模式抽取搜索结果中的并列词项. 不同的是，该方法通过监督学习，首先训练模型用于判断候选词是否为分面词项，然后训练模型用于判断两个候选词项是否属于同一个分面. QDMiner、QF-I、QF-J 是目前查询分面挖掘的最优方法. 如前所述，这几种方法主要是基于查询的初始搜索结果集来挖掘分面，因此分面的准确性和覆盖率很大程度上受限于查询结果集的质量，可能无法充分挖掘分面中应该具有的分面词项. 本文拟通过从大规模网页集合中挖掘出的词项关系信息来对查询分面进行扩充和完善.

在工业界，鉴于查询分面对提高用户满意度尤其是复杂查询和探索式查询下用户满意度的作用，目前各大商业搜索引擎也逐渐在搜索结果页面中插入类似分面的信息来帮助用户. 图 2 显示了 2015 年 8 月份在微软必应搜索引擎中查询“手机”时，显示在简单结果右侧的相关搜索信息. 和传统的查询建议列表不同，这些信息也采用了类似查询分面的结构化组织方式，即将相关手机型号和品牌等信息分组显示，便于用户浏览. 和本文中挖掘的查询分面不同，目前搜索引擎中的类似分面信息主要基于知识库和查询日志生成，分面数目、查询以及查询分面词项的覆盖度等还非常有限.

图 2　在必应中查询“手机”时显示的扩展信息

2.2　集合扩展

集合扩充问题指的是将一个包含少量词项的列表扩充成更完整的列表，例如将“C，C++，Java”扩展为所有编程语言. 目前主要的方法 SEAL[5]，利用搜索引擎返回的结果和种子，学习一个抽取器(wrapper)，并用它抽取同页面中具有类似模式的其他词项作为候选者，最后通过图算法将候选者排序，补充到原集合中. SEAL 对于种子的个数非常敏感，尤其当个数大于 5 个时，效果下降得很快，因为同时包含很多种子的文档很少. Wang 等人[6]提出了改进方法 iSEAL. iSEAL 多次调用 SEAL 过程，每次迭代选择少量的种子，同时 iSEAL 可以达到类似 bootstrap 的效果，使用极少的人工选取的种子作为初始，将每次迭代输出的高质量的未监督的词项作为下一次迭代的种子，在弱监督的情况下达到很好的效果. 集合扩充问题和本文探讨的查询分面扩展问题有相似之处，目的都是将一组同类词扩充得更完整，但也有许多不同. 查询分面有查询和搜索引擎返回的文档作为上下文，算法可以通过有效利用上下文来保证扩展的精度. 例如列表“周迅，刘德华，葛优”，如果仅是集合扩充问题，扩充的结果可能是中国的优秀演员；如果作为“冯小刚”这个查询的分面，扩展的目标就是和冯小刚合作过的演员，这对精度提出了更高的要求.

2.3　查询推荐

查询推荐是搜索引擎中广泛使用的技术之一. 和查询分面类似，查询推荐也是为了解决用户查询目标不明确、查询有歧义、对初始查询内容不满意的问题. 目前的查询推荐挖掘方法主要是基于查询反馈和查询日志分析等技术[7-11]. 目前在搜索引擎中广泛使用的查询推荐方法主要是基于查询频度的方法，倾向于返回和原查询语义相近或者更具体的高

频相关查询或扩展查询. 查询推荐中,返回的查询仅以列表形式展现,其中包含的各查询之间没有必然联系. 与此不同,构建查询分面的主要目的是对查询所能够涵盖的内容和知识进行总结. 查询分面是有结构的,每个分面内包含的词项在语义上是相关的或者对等的.

3 查询分面挖掘

如前所述,现有的查询分面挖掘方法 QDMiner[2]、QF-I 和 QF-J[3] 主要是通过挖掘查询的初始搜索结果集中包含的频繁列表来进行. 分面的准确性和覆盖率很大程度上受限于查询结果集以及其中包含的列表的数量和质量. 若查询的某类信息不以列表的形式或其他任何形式在搜索结果中展现,就无法挖掘出对应的分面信息或者导致挖掘到的分面中词项残缺不全.

针对这一问题,本文提出一种新的基于大规模词项图挖掘查询分面的算法. 整个算法的框架如图 3 所示. 其中的关键环节介绍如下.

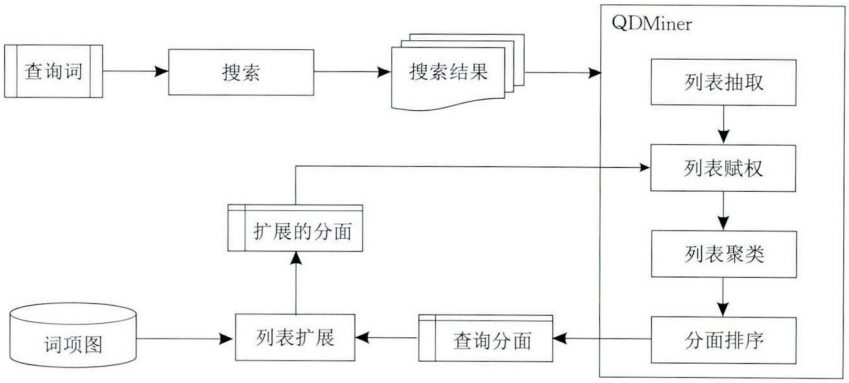

图 3 $\text{QDMiner}_{\text{TG}}$ 算法流程图

(1) 词项图构建. 挖掘大规模互联网网页语料中出现的并列或者平行信息,构建和查询无关的全局词项图. 在该词项图中,如果两个词有并列或者对等关系,则这两个词之间的距离较近.

(2) QDMiner. 给定查询,采用已有的查询分面挖掘算法 QDMiner[2] 对检索结果进行列表抽取、列表赋权、列表聚类以及分面排序等步骤,并生成初始查询分面.

(3) 列表扩展. 对于生成的查询分面集中的每一个查询分面,通过大规模全局词项图进行扩充,找到和原始词项相关的新词项作为补充生成新的查询分面. 这一过程中还可能会对原始词项的权重进行修改或者删除部分词项. 采用大规模全局词项图可以有效解决有限数量的检索结果中包含的列表以及词项有限的问题.

(4) 将新扩展的分面作为内容列表与原搜索结果中抽取出的内容列表合并后,重新使用 QDMiner 进行列表赋权、聚类、分面排序并生成新的查询分面.

(5) 重复(3)和(4),迭代地进行列表扩展和分面生成,直至查询分面结果稳定.

值得说明的是,本文主要从查询分面生成算法的效果方面进行研究和评测,高性能的分布式查询分面生成系统超出了本文的研究范围,我们将在未来的工作中对这一问题进行深入研究.

本文第 3.1 节介绍本文使用的大规模全局词项图是如何构建的;第 3.2 节简要介绍现有的 QDMiner 算法;第 3.3 节是本文的核心,介绍如何使用词项图扩展 QDMiner 生成的初始查询分面;最后,介绍如何将扩展后的分面和检索结果中抽取的并列词项列表融合在一起,来对上述过程进行迭代并生成最终的查询分面.

3.1 大规模词项图构建

本文基于史树明等人提出的从大规模互联网网页中挖掘语义类的方法[12-13] 来生成大规模词项图. 首先采用预定义抽取模式(如表 2 所示),从网页中抽取出原始平行词项列表及类别(每个平行词项列表可以看作一个原始语义类),然后基于原始语义类构建词项图(Term Graph)$G(V,E)$. 所有原始语义类中的词项所构成的集合即为图 G 的节点集合 V,每一对节点间都存在边(即该图可以看作完全图),边的权值是两个词项的相似性,或是对等词项的可能性. 实际上,如果两个词项从未在相同的原始语义类中共现,边的权值即为 0,等价于这条边不存在,把 G 看作完全图是为了讨论方便. 词项 a 和 b 的相

似性，即边的权重，按式(1)计算：

$$sim(a,b)=\sum_{i=1}^{m}\log\left(1+\sum_{j=1}^{k_i}w(P(C_{i,j}))\right) \quad (1)$$

其中：$C_{i,j}$是通过平行模式抽取出的同时包含词项 a 和 b 的一个原始语义类；$P(C_{i,j})$是用于抽取 $C_{i,j}$的的平行模式而 $w(P)$是该模式的权重. 假设同时包含 a 和 b 的原始列表来自于 m 个不同网站，k_i是第 i 个网站中包含的该类原始语义类的数目. 简单讲，如果两个词项在互联网上多次被同时并列提及，则他们的相似性或对等性较高，否则较低. 本文采用上述方法，基于 ClueWeb09[①] 数据集来构建大规模词项图. 词项图中的节点大部分是名词，但也包含其他词性的词或者短语. 如果一系列形容词或动词满足表 2 中的模式，这些词也会被加入到词项图中.

表 2　词项图构建中使用的平行词项挖掘模式

类型	抽取模式
文本	NP {, NP *}{,}(and|or){other}NP
网页标记	〈UL〉〈LI〉item〈/LI〉…〈LI〉item〈/LI〉〈/UL〉 〈SELECT〉〈OP〉i…〈/OP〉i〈/SELECT〉

需要指出的是，词项图与知识图谱是有区别的. 两者虽然都包含词或者词项之间的关联关系，但词项图是从大规模互联网中抽取词项构建的，而知识图谱中存储的主要是物理世界存在的实体，所以词项图的涵盖面更全. 知识图谱的优势在于信息质量高、杂质少、关系表达更为精确. 如何使用知识图谱或知识库生成查询分面也是我们未来研究的方向. 除此之外，尝试在其他的数据集上、或者使用不同的方法构建词项图，也是我们未来工作的内容.

可以看出，上述词项图的构建过程中所基于的基本思想和采用的平行词项抽取模式和 QDMiner[2] 有类似之处，核心任务都是找出词项之间的对等性和相似性. 不同之处在于，QDMiner 采用了查询的搜索结果来对词项进行聚类并生成查询相关的分面；在生成分面过程中，重点完成对词项列表的聚类，不直接计算词项之间的相似性. 而大规模词项图基于大规模网页集合构建，不针对具体查询，包含的词项数量和覆盖面要高于 QDMiner. 此外，词项图在生成时直接计算词项之间的相似性而不强制对初始列表进行聚类，保留了更多关于词项关系的信息. 因此，通过使用大规模词项图，有潜力能够进一步改进现有分面挖掘方法的性能：一方面引入更多对等和相似词项，提高原分面的召回率；同时可以通过大规模词项图中的词项相似性，剔除原始分面中包含的噪声或者垃圾词项，提高词项准确度.

3.2　基准查询分面生成算法 QDMiner

本文采用 QDMiner 算法[2] 挖掘初始查询分面. 给定查询 q，从搜索引擎获取前 K 个检索结果并将这些结果的内容抓取下来，组成查询结果集 R. 本文使用 Bing(必应)搜索引擎获得查询结果. 下面将简要介绍一下 QDMiner 算法的核心步骤. 如图 3 所示，QDMiner 通过下述 4 个处理过程从 R 中挖掘出查询分面.

(1) 列表抽取. 基于文本、HTML 标签、重复区域等模式，从 R 中的每一个文档中抽取列表. 基于 HTML 标签重点抽取下拉列表、复选框、表格等 HTML 标签中存在的并列结构. 从剔除 HTML 标签的纯文本中，重点抽取并列短语. 例如在文本“北京、上海、天津和重庆是中国的直辖市”中，可抽取出{北京，上海，天津，重庆}这个列表. 除这两种模式外，还进一步对网页中包含的重复区域进行检测，然后抽取重复区域中每一条记录中的对应属性作为一个列表. 相似的抽取算法请见文献[2]，本文不再赘述.

(2) 列表赋权. 评估抽取出来的每一个列表的重要性，并将重复的列表进行合并. 在衡量列表重要性时，利用列表中每一个词项在全部检索结果中出现的频度. 若一个列表中的每个词项都在搜索结果中出现很多次，同时在大规模互联网语料中出现的不多，则该列表倾向于是一个重要列表. 经过赋权后，和查询相关的有重要意义的列表倾向于被赋予较高的权重，而垃圾、无用或者高频词构成的列表则权重较低.

(3) 列表聚类. 将相似的列表合并在一起形成查询分面. 在聚类时，如果两个列表包含很多相同词项，则可能会被聚在一起. 最终生成的每一个聚类即为一个查询分面. 在 QDMiner 算法中采用了 WQT 算法进行聚类.

(4) 分面排序. 对前 3 步生成的查询分面的重要度进行评估，并按照计算出的重要度将分面进行排序并输出. 一般来说，如果一个分面包含的原始列表较多，来源网站重要且数目较大，则该分面的重要性较高；相反则较低. 若一个词项包含在较多的查询列表中并且经常出现在列表中的靠前位置，则该词项在最终查询分面中的位置也较靠前.

从上述 4 个步骤可以看出，QDMiner 算法主要是通过挖掘查询的初始搜索结果集中包含的频繁列

① ClueWeb09 dataset. http://www.lemurproject.org/clueweb09/

表来生成查询分面.分面的准确性和覆盖率很大程度上受限于查询结果集以及其中包含的列表的数量和质量.若查询的某类信息不以列表的形式或其他任何形式在搜索结果中展现,则在第(1)步列表抽取时便无法抽取相关词项,进而也不可能生成包含该词项的相关查询分面,导致最终挖掘到的分面中包含的词项不全,或者缺失某些重要的查询分面.针对这一问题,在下一节中将介绍如何利用前文介绍的从大规模网页中挖掘出的词项之间的关系,对初始分面进行扩充,进而生成更完整、更准确的查询分面.

3.3 基于分面扩展的分面挖掘算法 $\text{QDMiner}_{\text{TG}}$

如图3所示,本文的核心是对QDMiner算法挖掘出的初始查询分面进行扩展,进而生成准确率和召回率更高的查询分面.查询分面的扩展算法主要由以下几个步骤构成:

(1)获得待扩展列表集合 L;

(2)对于 L 中的每个列表 l,基于词项图对列表进行扩展,生成候选词项集合 $C(l)$;

(3)对于 $C(l)$ 中的每个词项 c,计算不同类型的特征,用于评估该词项是否可以被扩展到原分面或列表中.具体的特征所代表的含义以及计算方法将在下文进行详细介绍;

(4)采用SVM分类器,预测每个候选词项是否应加入到分面中.所有预测为正例的候选词项构成新的列表 l_{new};

(5)对 l_{new} 重新应用上述步骤,进行迭代直至列表中的分面项不再发生变化后输出.最终的分面排序方法和QDMiner相同.

在本节的剩余内容中将分别对上述步骤进行详细介绍.

3.3.1 待扩展列表选取

待扩展列表来源有两种:一是直接从搜索结果中抽取的列表,即QDMiner算法第1步"列表抽取"的输出.主要为抽取出的网页的文本、HTML内容和重复区域中包含的列表.按照Dou等人的统计,每个查询的前100个结果中可以抽取近千个列表[2].二是QDMiner算法通过赋权、聚类、排序生成的查询分面.这种查询分面中包含的所有分面项组成了待扩展列表.这种列表因为经过了查询分面算法的处理,质量更高.本文仅对第2种列表,即初始分面进行扩展.对于原始列表的扩展研究将在后续工作中进行.

3.3.2 候选词项获取

对于每个列表 l,使用词项关系图 G 中距离待扩展列表包含的词项的邻近节点作为候选扩展词项.对于 l 中的每个词项 e,取和 e 距离最小的前 p 个节点(在本文中,取 $p=100$)作为候选加入整个列表的候选词项集合 C,即

$$C=\bigcup_{\forall e\in l} n_p(e) \tag{2}$$

其中 $n_p(e)$ 代表 G 中距离词项 e 最近的前 p 个词项.词项之间的距离计算如式(1)所示.在这一步中,有可能会引入不包含在原来的检索结果或者列表中的词项,这是能够提高最终分面的质量尤其是分面项的召回率的最主要原因.

3.3.3 候选词项重要性特征

选取出的候选扩展词项可能有上千个.这些候选词项和原始分面或者待扩展列表中包含的词项的相似性不一.某些候选词项与原始分面或者列表的语义可能会出现偏离.在本文中,采用分类的方法对候选词项能否作为最终分面项进行判断和预测.下文中将提出对扩展词项的重要性进行评估的一系列特征,这些特征将用于训练分面项分类器.

(1)基于词项子图的特征 $w_{\text{graph}}(c)$

假设原始待扩展列表中的词项的质量较高,若新检索到的候选词项和每个带扩展列表中的词项的相似性都很高,则认为该候选词项与原始列表的语义相关性较高,被判定为最终分面项的几率也较大.基于这个思想,采用基于图的节点重要性判定方法来对候选词项的重要性进行判定.候选词项集合 C 是图 G 中节点的子集.基于 C 构建 G 的子图,使用随机游走算法可以找到与列表 l 最普遍相关的候选词项.类似PageRank算法,本文采用的随机漫步算法首先对 l 中的每个词项 e 赋予一个初始值(在本文中将所有在 l 中出现的词项的初始值设置为1).随后的每次迭代中,每个节点的值都会随机地流到某个与之相邻的节点,流动的概率与两个节点关联的强弱相关,从源节点 x 到相邻节点 y 的概率可以表示为

$$P(y\mid x)=\frac{sim(x,y)}{\sum_{\forall x'\in n_p(x)} sim(x,x')} \tag{3}$$

为了避免死循环,在每一步迭代中,有 d 的概率不发生值的流动(本文设置 $d=0.15$).随机漫步会一直进行下去直到图中节点的值趋于稳定,也就是相邻两次迭代的节点值的差值平方和小于某个阈值(本文设为0.01).经过观察发现一般经过5~8

次迭代，图中的值就趋于稳定. 最终采用节点的值作为对应候选词项的重要性 $w_{\text{graph}}(c)$，该值本质上代表了每个候选词项和原始列表中种子词项的相似度. 一般来说 $w_{\text{graph}}(c)$ 的值越大，c 为分面项的可能性越大.

(2) 检索结果匹配程度 $w_{\text{content}}(c)$ 和 $w_{\text{list}}(c)$

词项图与搜索引擎返回的结果有很大的不同. 词项图中包含的词项有可能更全面、丰富，但是某些候选词项可能与原始分面或者列表的语义差异较大. 文本特征试图将搜索引擎检索结果结合进来，从而避免在使用语料库的情况下过分地扩大用户的查询意图. 例如查询“迈克尔杰克逊”的一个分面是他的歌曲. 在语料库中，甲壳虫乐队的歌曲“Yesterday”和迈克尔杰克逊的歌曲在很多页面中共同出现，所以基于图的特征会把“Yesterday”作为杰克逊歌曲的普遍相关的节点，从而错误地扩大了原分面所要涵盖的范围.

因此，每个候选词项都需要通过搜索引擎结果的进一步赋权. 文本匹配特征衡量每个候选者在搜索到的文档中的出现情况. 本文采用两个特征来刻画候选词项与检索结果的匹配程度：一是每个候选者在文档中出现的次数；二是每个候选者和原列表 l 共同出现的次数.

如前所述，设 R 是检索结果文档集合，$d \in R$ 是一个文档，$rank_d$ 是文档 d 在集合 R 中的排序位置. 候选词项在检索结果集中的匹配程度 w_{content} 计算如下：

$$w_{\text{content}}(c) = \sum_{\forall d \in R, c \in d} \frac{1}{\log_{10}(10 + rank_d)} \quad (4)$$

其中 $c \in d$ 代表词项 c 出现在文档 d 中.

候选词项和原列表 l 中的词项共现情况 w_{list} 按如下方式计算：

$$w_{\text{list}}(c) = \sum_{\exists e \in l, c \in d, e \in d, \forall d \in R} \frac{1}{\log_{10}(10 + rank_d)} \quad (5)$$

因为 w_{content} 和 w_{list} 从不同角度上衡量了候选词项与检索结果匹配的程度，本文同时使用 w_{content} 和 w_{list} 作为候选词项的特征.

(3) 种子词项重要性 $w_{\text{seed}}(c)$

由于噪声的存在，种子列表的质量可能并不好. 例如查询“Apple”的列表“Google，Yahoo，Microsoft，See More，…”中“See More”明显是噪声项. 消除这种噪声的方法有很多种，例如一般“See More”、“Next”等噪声项的超链接特征和正常项链接是不一样的，可以将这些特征作为判断的依据，除此之外还可以衡量列表中每个词项的重要性. 对于列表 l 中的词项 e，首先考虑该词项在 G 中的关联候选词项的重要性. 若 e 的相关词项和 e 关系紧密且和 l 中其他词项也相关，则 e 的重要性也较高；否则 e 可能为噪声词项. 基于此思想，采用下述公式计算 e 的重要性：

$$importance(e) = \sum_{\forall c \in n_p(e)} w_{\text{ove}}(c) \cdot sim(e, c) \quad (6)$$

其中，$w_{\text{ove}}(c)$ 为候选词项在 l 中种子词项的邻居中出现的次数，即

$$w_{\text{ove}}(c) = \frac{|\{e | c \in n_p(e), \forall e \in l\}| - 1}{|l|} \quad (7)$$

同时，还需要考虑 e 在初始查询分面或列表中的位置. 若 e 的位置靠前，则重要性较高. 因此，词项 e 的最终重要程度可计算为

$$w_{\text{item}}(e) = \frac{\log_{10}(10 + importance(e))}{\log_{10}(10 + rank(e|l))} \quad (8)$$

其中，$rank(e|l)$ 是 e 在列表 l 中的位置.

在计算了每个种子词项的重要性后，对候选词项 c，可反过来计算它和种子词项的相似度并同时考虑每个种子词项的重要度，即

$$w_{\text{seed}}(c) = \sum_{e \in n_p(c), \forall e \in l} w_{\text{item}}(e) \cdot sim(c, e) \quad (9)$$

(4) 候选词项同质性 $w_{\text{hom-idf}}(c)$ 和 $w_{\text{hom-len}}(c)$

为了保证扩展后的列表的同质性，计算每个候选词项的 IDF (Inverse Document Frequency). 假设列表中每个词项在互联网中出现的频率都应该比较接近. 因此，若某个候选词项的 IDF 值和种子词项的平均 IDF 相差太远，则其重要性越低. IDF 的计算公式如下：

$$idf(c) = \log \frac{N - N_e + 0.5}{N_e + 0.5} \quad (10)$$

其中：N_e 表示语料库中包含词项 e 的文档的个数；N 表示语料库中文档总数. 本文使用 ClueWeb09 数据集作为计算 IDF 的语料库.

$$w_{\text{hom-idf}}(c) = \frac{2}{1 + e^{-|idf(c) - avgidf(l)|}} - 1 \quad (11)$$

其中 $avgidf(l)$ 为 l 中所有原始词项即种子词项的 IDF 的平均值. 列表同质性的另外一个特征为词项的长度，如果候选词项和种子词项的长度差异太大，则有可能不是一个合法的词项，即

$$w_{\text{hom-len}}(c) = \frac{2}{1 + e^{-|len(c) - avglen(l)|}} - 1 \quad (12)$$

其中 $len(c)$ 为词项 c 的长度，而 $avglen(l)$ 为 l 中词项的平均长度.

3.3.4 扩展词项选定及扩展词项生成

上一节中介绍了各种候选词项重要性评估特征 $w_{\text{graph}}(c)$、$w_{\text{content}}(c)$、$w_{\text{list}}(c)$、$w_{\text{seed}}(c)$、$w_{\text{hom-idf}}(c)$和 $w_{\text{hom-len}}(c)$. 基于上述特征,本文使用 SVM 分类器并采用 RBF 核函数来训练扩展词的分类模型. 该模型用于预测一个词项是否可以选定为扩展词项来扩展原始列表. 最终将被分类器预测为可扩展项的词项选定,并按照分类器输出的属于正例的概率值对选定词项进行排序并输出. 因为种子词项也已经全部包含在候选列表中,因此输出的列表即为最终扩展列表而不需要在和原始输入列表进行融合.

3.3.5 分面生成及迭代

经过以上过程,可得到原列表 l 的扩展列表 l_{new}. 此时采用 QDMiner 系统,从第 2 步开始重新进行列表赋权、列表聚类、分面排序等步骤,最终生成新的查询分面. 对于得到的分面可继续采用上述方式进行列表扩展并重新生成分面. 该过程可重复迭代多次,直至分面和分面词项内容稳定,即没有新分面和词项出现且各词项在分面中的位置不再发生变化. 在实际实验中,发现大部分查询平均迭代 2～4 次后就基本生成稳定的分面结果. 具体实验结果和分析在下一节中进行介绍.

4 实验与分析

4.1 实验设置与评价指标

本文采用文献[2]中使用的两个数据集对提出的算法进行实验验证. 其中,数据集 *UserQ* 由 89 个查询组成,查询主要来源于 QDMiner 用户. *RandQ* 是在必应搜索引擎查询日志中进行随机抽样获得的 105 个查询. 对于每个查询,采用人工标注的方法手动生成了查询分面. 对于每个查询分面,进一步标注了三级重要度 Good(高质量)/Fair(一般质量)/Bad(不相关). 其中每个查询分面的重要度由至少 5 个标注者进行标注,最终的分面重要度为得票最多的标注等级.

经过人工标注,平均每个 *UserQ* 中的查询分别有 4.9、5.3、4.4 个 Good、Fair、Bad 级别的查询分面,平均每个 *RandQ* 中的查询分别有 2.9、2.1、2.1 个 Good、Fair、Bad 级别的查询分面. *UserQ* 中的查询的分面个数明显多于 *RandQ* 中的查询,原因是 *RandQ* 中的查询是从日志中随机抽取的,一些噪声查询或过于明确的查询不存在有意义的分面. 数据集的其他一些统计特征请见表 3.

表 3 实验数据统计信息

描述	*UserQ*	*RandQ*
查询数	89.0	105.0
平均每个查询的结果数	99.8	99.5
平均每个文档的列表数	44.1	37.0
平均每个列表的词项数	97.0	10.1
平均每个查询的标注分面数	10.2	4.2

本文采用文献[2]提出的 $fp\text{-}nDCG$ 和 $rp\text{-}nDCG$ 评价最终查询分面的质量. 其中 $fp\text{-}nDCG$ 是在 $nDCG$[14] 的基础上融入了每个分面词项的准确率,具体计算公式如下:

$$fp\text{-}nDCG_p = \frac{\sum_{i=1}^{p} fw_i \cdot DG_i}{IDCG} \tag{13}$$

其中 $fw_i = \frac{|c_i \cap c_i^m|}{|c_i|}$.

可以看出,$fp\text{-}nDCG$ 融入了分面的排序质量和聚类的 $Purity$ 值. $rp\text{-}nDCG$ 在 $fp\text{-}nDCG$ 的基础上又融入了分面词项的召回率,具体计算公式如下:

$$rp\text{-}nDCG_p = \frac{\sum_{i=1}^{p} rw_i \cdot DG_i}{IDCG} \tag{14}$$

其中 $rw_i = \frac{|c_i \cap c_i^m|}{|c_i|} \cdot \frac{|c_i \cap c_i^m|}{|c_i^m|}$.

除上述指标外,本文还使用了孔维泽等人[3]提出的 PRF 和 $wPRF$ 指标. 其中 PRF 指标融合了词项的准确度、召回率和聚类 $F1$,具体计算公式如下:

$$PRF(c_i, c_i^m) = \frac{(\alpha^2+\beta^2+1)prf}{\alpha^2 rf + \beta^2 pf + pr} \tag{15}$$

其中 p,r,f 分别是分面词项的准确率、查全率、聚类 $F1$. 而 $wPRF$ 融合词项的加权准确度、召回率和加权聚类 $F1$. 具体计算公式为

$$wPRF(c_i, c_i^m) = \frac{(\alpha^2+\beta^2+1)p_w r_w f_w}{\alpha^2 r_w f_w + \beta^2 p_w f_w + p_w r_w} \tag{16}$$

其中 p_w,r_w,f_w 是考虑不同词项的权重后的分面的准确率、查全率和聚类 $F1$.

在上面公式中,c_i 是算法生成的第 i 个分面,而 c_m^i 是第 i 个分面所对应的标注结果中的基准分面. $fp\text{-}nDCG$ 考虑了分面的准确率,而 $rp\text{-}nDCG$ 既考虑了准确性,又考虑了查全率. PRF 和 $wPRF$ 的缺陷是仅考虑了分面项的质量,包括准确率、查询率、聚类质量等,却没有考虑分面的排序质量(即每个分面和查询的相关性质量). 如果系统输出的分面内容一致,则不管哪个分面放在输出列表前面,用 PRF

和 $wPRF$ 评价时，都是无法区分的. 事实上，在很多应用中，分面质量和有用性是至关重要的. 例如，当查询为“手表”时，关于手表品牌的分面的有用性要远远高于手表颜色. 因此在评价时，$fp\text{-}nDCG$ 和 $rp\text{-}nDCG$ 的价值高于 PRF 和 $wPRF$.

本文对每个查询使用微软必应搜索引擎①抓取的前 100 个搜索结果进行初始查询挖掘，使用 ClueWeb09(参见本文第 6 页脚注①)数据集挖掘语义类并建立词项关系图. 本文实现了目前最好的分面挖掘算法 QDMiner 算法[2]和 QF-I、QF-J 算法[3]作为基准方法. 本文提出的算法命名为 $\text{QDMiner}_{\text{TG}}$. 对于 $\text{QDMiner}_{\text{TG}}$、QDMiner 和 QF-I 算法，都使用 5 折交叉验证来进行参数模型训练和评测. 对于 QDMiner，采用交叉验证来选择列表聚类的两个参数：聚类直径 $Dia_{\max}$ 和最小聚类权重 $W_{\min}$. 对于 $\text{QDMiner}_{\text{TG}}$，除了选择上面两个参数外，还通过交叉验证训练 SVM 分类模型. 对于 QF-I 算法，训练聚类权重以及分面项分类阈值. QF-J 中不包含任何可调节的参数，不需要进行交叉验证. 本文仅评价返回的前 10 个查询分面. 为了简单起见，下文将忽略评价指标中的分面数目标识，即用 $fp\text{-}nDCG$ 表示 $fp\text{-}nDCG_{10}$，用 $rp\text{-}nDCG$ 代表 $rp\text{-}nDCG_{10}$. PRF 和 $wPRF$ 也均在前 10 个分面上进行计算.

4.2　扩展词项分类

在采用 SVM 分类器训练词项分类模型时，采用数据集中标注为 Good 或者 Fair 的分面中包含的词项作为正例，并在候选词项中随机选取等数量的其他词项作为负例. 在进行交叉验证时，SVM 分类模型在各组交叉验证中的训练数据上进行训练，并应用在测试数据上. 在 5 折交叉验证后，在 $UserQ$ 上 SVM 分类器的准确率($Precision$)和召回率($Recall$)分别为 0.91 和 0.75，在 $RandQ$ 上精确度为 0.85，召回率为 0.74.

表 4 中显示了更详细的关于每个特征的分类效果数据. 在该表中，“全部特征”表示在训练和使用 SVM 模型时，使用上文介绍的全部特征. 而表中的“-特征”表示在特征集合去除掉某一特征后的效果. 例如“$-w_{\text{graph}}$”表示去除 w_{graph} 这一特征后的分类效果. 如果去除某一特征后分类性能明显下降，则表明该特征和其他特征之间的冗余较小，对分类贡献较大. 反之，如果去除某一特征后分类效果没有明显变化，则意味着这一特征的作用不大. 表 4 显示在 $UserQ$ 数据集上，影响最大的 3 个特征分别为 w_{content}、w_{graph} 和 w_{seed}. 其中去除 w_{content} 后，分类的召回率($Recall$)增加，但准确率($Precision$)大幅下降. 这是因为使用词项图很有可能导致分面或者待扩展列表的语义范围扩大. 例如，查询“张艺谋”中有一个分面包含的内容为张艺谋导演过的电影. 在使用词项图时我们发现很多其他电影，例如某一年的热门电影，即使不是张艺谋导演的，也和张艺谋导演的电影的距离较近. 这可能是因为生成该词项图的大规模网页中包含了各个方面的信息(例如每年的获奖电影列表)，导致在使用该词项图时，语义距离较近的词项并不一定是属于原始分面的. 使用检索结果的内容可以有效控制一些语义差距较大的词项的出现. 通常，在查询“张艺谋”的检索结果中出现的电影是张艺谋导演的可能性要远远高于其他电影. 因此，使用 w_{content} 特征可以大幅度提高分类的准确率(从 0.8 提升到 0.91). 因为个别分面项不出现在检索结果中，使用该特征的负面影响是会导致部分不出现在检索结果中的分面项无法识别. 但因为在使用该特征时并不直接删除那些不在结果中出现或者出现次数很少的分面项，因此该特征对 $Recall$ 的影响不大. 表 4 显示在使用该特征后，$Recall$ 从 0.77 降到了 0.76，但因为准确率的提高，最终的 $F1$ 提高了 0.044. 特征 w_{list} 的作用类似，但因为该特征仅考虑和分面项在列表中同时出现的情况，影响的分面项的数量远远低于 w_{content}，因此在 $UserQ$ 上该特征的作用不如 w_{content} 明显. 使用 $W_{\text{hom-idf}}$ 和 $W_{\text{hom-len}}$ 这两个特征可以小幅提高结果准确率，对召回率的影响不大. w_{graph} 和 w_{seed} 特征在 $UserQ$ 和 $RandQ$ 上的作用

表 4　候选词项分类中各特征的作用(每个特征对应的分类效果为从所有特征中去除该特征后重新训练后的分类效果)

特征	$UserQ$			$RandQ$		
	$Precision$	$Recall$	$F1$	$Precision$	$Recall$	$F1$
全部特征	0.91	0.76	0.83	0.86	0.74	0.80
$-w_{\text{graph}}$	0.86	0.75	0.80(−0.027)	0.81	0.72	0.76(−0.033)
$-w_{\text{content}}$	0.80	0.77	0.78(−0.044)	0.79	0.76	0.77(−0.021)
$-w_{\text{list}}$	0.84	0.79	0.81(−0.014)	0.80	0.75	0.77(−0.021)
$-w_{\text{seed}}$	0.88	0.74	0.80(−0.024)	0.84	0.73	0.78(−0.014)
$-w_{\text{hom-idf}}$	0.90	0.75	0.82(−0.010)	0.85	0.74	0.79(−0.004)
$-w_{\text{hom-len}}$	0.89	0.76	0.82(−0.008)	0.84	0.74	0.79(−0.009)

① Bing Search Engine, http://www.bing.com/

都很明显,证明采用词项图中的词项本身的特征和词项之间的关系是非常有效的.

4.3 分面质量

表 5 分别显示了各种方法以不同指标为基准指标进行交叉验证后的查询分面质量.从图中可以看出:

(1)本文提出的新算法 $\text{QDMiner}_{\text{TG}}$ 比基准方法 QDMiner 的 $fp\text{-}nDCG$ 指标略高.而在 $rp\text{-}nDCG$ 指标上,$\text{QDMiner}_{\text{TG}}$ 算法明显优于 QDMiner.对两者的 Student 双边 T 检验显示 $p<0.01$,说明 $\text{QDMiner}_{\text{TG}}$ 对 QDMiner 的改进是显著的.经过观察发现,QDMiner 可以显著提高查询分面中挖掘到的词项的数目,这是带来 $rp\text{-}nDCG$ 大幅提升的最主要原因.同时 $fp\text{-}nDCG$ 值略有提高说明新扩展的词项的质量较高,没有引起整体分面准确性的下降.经过观察详细实验数据我们发现 $\text{QDMiner}_{\text{TG}}$ 还可以删除原始查询分面中的部分噪声词项,带来 $fp\text{-}nDCG$ 值的小幅提升.

(2)各种算法结果的召回率($rp\text{-}nDCG$)的分值都明显低于准确率($fp\text{-}nDCG$),这说明目前这些算法的返回的分面项的覆盖率虽然明显提高,但整体仍然还有较大的改进空间.造成这一结果的原因主要有以下几点:首先,有些分面词项并不在返回的排序靠前的文档集合中,或者出现在文档中但现有的抽取模式很难提取.这是本文尝试解决的问题.如上所述,表 5 显示本文提出的算法 $\text{QDMiner}_{\text{TG}}$ 的 $rp\text{-}nDCG$ 分值已经明显高于基准方法,这说明使用大规模词项图已经能够在一定程度上解决上述问题.但因为在使用词项图时,很多查询相关的词项如果在词项图中出现次数较少,置信度较低,也无法在扩展词项时检索到.其次,全面的获取一个分面的所有词项也是没有必要的.例如对于查询"手表",一个分面是"手表品牌".手表的品牌可能有很多.本文的实验数据显示标注集中手表的品牌多达 100 多个.对于用户而言,最关心的其实是这些品牌中的知名品牌,在分面中包含小众的品牌意义并不大.

(3)在 PRF 指标(表 6)和 $wPRF$ 指标(表 7)上,$\text{QDMiner}_{\text{TG}}$ 也显著优于基准方法 QDMiner,说明词项的质量有改善.大量高质量新词项的引入也是这两个评价指标能有提高的本质原因.

表 5 分面质量

(a) $fp\text{-}nDCG$ 结果(基于 $fp\text{-}nDCG$ 做交叉验证)

	QDMiner	QF-I	QF-J	$\text{QDMiner}_{\text{TG}}$
UserQ	0.6314	0.4687	0.3999	0.6409
RandQ	0.6400	0.4167	0.3482	0.6515

(b) $rp\text{-}nDCG$ 结果(基于 $rp\text{-}nDCG$ 做交叉验证)

	QDMiner	QF-I	QF-J	$\text{QDMiner}_{\text{TG}}$
UserQ	0.2099	0.1556	0.1350	0.2370
RandQ	0.2618	0.1701	0.1289	0.3012

(c) PRF 结果比较(基于 PRF 做交叉验证)

	QDMiner	QF-I	QF-J	$\text{QDMiner}_{\text{TG}}$
UserQ	0.3603	0.4329	0.3426	0.4210
RandQ	0.3853	0.3730	0.2470	0.3966

(d) $wPRF$ 结果比较(基于 $wPRF$ 做交叉验证)

	QDMiner	QF-I	QF-J	$\text{QDMiner}_{\text{TG}}$
UserQ	0.3725	0.4726	0.3630	0.4580
RandQ	0.4090	0.4155	0.2770	0.4186

表 6 $\text{QDMiner}_{\text{TG}}$ 算法迭代次数统计(实际扩展次数为算法能够输出最终结果的迭代次数.例如,n 次扩展代表第 $n-1$ 次运行的结果与第 n 次的结果相同)

实际扩展次数	分面个数	
	UserQ	*RandQ*
2 次	1028(36%)	930(41%)
3 次	1257(44%)	839(37%)
4 次	400(14%)	365(16%)
5 次	86(3%)	95(4%)
6 次	29(1%)	34(1%)
大于 6 次	57(2%)	5(0.2%)
合计	2857	2268

表 7 $\text{QDMiner}_{\text{TG}}$ 算法在限定最大迭代次数后的性能

(a) 在 *UserQ* 数据集上的效果

最大扩展次数	*UserQ*			
	$fp\text{-}nDCG$	$rp\text{-}nDCG$	PRF	$wPRF$
2 次	0.6351	0.2183	0.3812	0.3865
3 次	0.6402	0.2235	0.3923	0.4023
4 次	0.6402	0.2286	0.4112	0.4423
5 次	0.6409	0.2362	0.4123	0.4502
6 次	0.6409	0.2370	0.4185	0.4550
不限制	0.6409	0.2370	0.4210	0.4580
QDMiner	0.6314	0.2099	0.3603	0.3725

(b) 在 *RandQ* 数据集上的效果

最大扩展次数	*RandQ*			
	$fp\text{-}nDCG$	$rp\text{-}nDCG$	PRF	$wPRF$
2 次	0.6405	0.2784	0.3901	0.4100
3 次	0.6483	0.2836	0.3954	0.4135
4 次	0.6490	0.2976	0.3954	0.4130
5 次	0.6515	0.3010	0.3966	0.4180
6 次	0.6515	0.3010	0.3966	0.4186
不限制	0.6515	0.3012	0.3966	0.4186
QDMiner	0.6400	0.2618	0.3853	0.4090

(4)本文提出的方法 $\text{QDMiner}_{\text{TG}}$ 在 $fp\text{-}nDCG$ 和 $rp\text{-}nDCG$ 两个指标上明显优于基准方法 QF-I 和 QF-J($p<0.01$).在 *UserQ* 数据集上,$\text{QDMiner}_{\text{TG}}$ 的 PRF 和 $wPRF$ 性能略低于 QF-I,而在 *RandQ* 上则略优于后者(上述差异都不是显著的).因为 PRF 和 $wPRF$ 只考虑了词项的准确率、召回率和聚类性

能，而没有考虑分面的排序性能，上述结果对比可说明 $\text{QDMiner}_{\text{TG}}$ 在词项准确性方面和 QF-I 接近，但排序质量远远高于 QF-I. 事实上，和搜索引擎中结果排序类似，分面的排序质量对用户体验的影响是很大的，因此在 QDMiner 和 $\text{QDMiner}_{\text{TG}}$ 算法设计中都倾向于将最优的分面排在分面列表顶部，而将其他分面排在后面.

(5) QF-J 在 *RandQ* 和 *UserQ* 上性能很差. 经过观察发现 QF-J 算法在进行后处理时，仅有少数词项可以被保留，导致了召回率等指标的低下.

4.4　算法性能分析

如第 3 节所述，本文重点关注的是分面算法的精度和效果. 由于涉及从搜索引擎中获取检索结果、列表抽取、列表聚类、列表扩展的迭代等时间消耗较大的操作，如果在线应用 $\text{QDMiner}_{\text{TG}}$ 以及现有的其他查询分面挖掘算法，则每个查询的时间代价都较高. 在本论文的实验中，每个查询的检索结果是提前抓取并存储在本地的，避免重复获取和每次运行时结果不一致. 大规模词项图已经提前生成好并提供在线查询. 在实验过程中在线进行列表抽取、聚类、初始分面生成、候选词项检索、迭代分面扩展等操作. 实验过程中，在单台 4 核 Intel Core i7-4790 CPU，24 GB 内存的 PC 机上，每个查询生成或者处理的分面个数约为 32 个，每个查询的算法运行时间约为 12 s.

虽然在本论文的实验中采用的是近似在线的分面生成方法，在实际应用中，应考虑算法的离线实现来提高算法效率. 一方面，热点查询的查询分面可以离线提前生成并缓存. 例如可以对最近一段时间的热门查询批量生成分面并将生成的分面结果进行缓存，用户在输入查询时可直接从缓存中直接获得查询分面. 同时，在查询分面挖掘时，可以使用一系列方法来提高分面生成效率. 例如，可以采用点击日志中出现的结果来代替检索结果集，这样就不需要每次获取查询结果；可以通过分布式系统如 Hadoop 等同时对多个查询进行处理，将以查询为单位的处理优化为一次同时处理多个查询的批处理；文档中列表抽取等步骤也可以在抓取到文档后统一进行处理并将抽取出的列表进行存储. 上面这些操作可大幅提高查询分面生成的效率. 本文主要从查询分面生成算法的效果方面进行研究和评测，高性能的分布式查询分面生成系统是本文的后续工作之一.

4.5　迭代次数分析

如前文介绍，$\text{QDMiner}_{\text{TG}}$ 可将 QDMiner 生成的每一个原始分面进行扩展并生成新的查询候选分面，然后对新得到的分面继续采用上述方式进行列表扩展并重新生成分面. 该过程可重复迭代多次，直至分面和分面词项内容稳定，即没有新分面生成或者分面中各词项的内容和位置不再发生变化. 本文对算法在实际运行时的迭代过程进行了分析，具体结果如表 6 所示. 该表显示算法在大多数分面上经过 2～4 轮即可结束迭代过程，输出稳定的分面结果. 在 *UserQ* 数据集上，约 94% 的分面都可以在 2～4 次迭代内输出稳定结果，而在 *UserQ* 上，约有 94.8% 的分面可以在 2～4 次内输出最终结果. 该表同时也表明多次迭代是有效果的，如果只进行一次运算，大概只有 40% 左右(在 *UserQ* 上为 36%，*RandQ* 上为 41%)的分面扩展结果和最终稳定的迭代结果内容相同.

在实际应用中，为了避免对某些分面的多次迭代带来较长的运行时间，可限制每个分面在扩展时的最大迭代次数. 表 7 中给出了在限定最大迭代次数的情况下，生成的查询分面的精度. 该图显示整体上限定的允许迭代次数越高，分面质量越高. 不管是在 *UserQ* 还是 *RandQ* 数据集上，在只允许进行 2 次迭代的情况下，$\text{QDMiner}_{\text{TG}}$ 也能在 QDMiner 的基础上，有效地改进分面质量. 当迭代次数接近 5 次时，输出的分面质量和不限制迭代次数的质量基本相同. 例如，在 *UserQ* 数据集上，$fp\text{-}nDCG$ 得分在最多允许 5 次迭代的情况下和不限制次数时的得分相同，$rp\text{-}nDCG$ 的得分(0.2362)也非常接近最终得分(0.2368). 这是因为按照表 6 所示，大部分分面都能够在 2～4 次内完成迭代.

在实际应用中，如果需要考虑分面生成的时间代价(即使是在离线处理中)，则可以根据上述实验结果考虑限定一定的最大扩展次数，在牺牲一部分分面质量的情况下提高时间效率. 同时，如前所示，因为在实际使用中，一般用户仅关注前几个分面，因此在进行分面扩展时没有必要对所有生成的初始分面进行扩展，可仅对排序较高的前几个分面应用 $\text{QDMiner}_{\text{TG}}$ 算法.

5　结　论

本文主要针对现有查询分面挖掘算法中存在的

仅依赖搜索结果、分面项召回率低的问题，提出了基于大规模互联网词项图来辅助搜索结果进行查询分面挖掘的方法. 方法的重点在于结合图、文本匹配等多种特征，利用词项图中提供的信息，循环地扩展、改善从搜索结果中挖掘到的初始查询分面. 本文使用公开的 *UserQ* 和 *RandQ* 作为测试数据评价算法效果. 实验结果表明使用基于大规模词项图进行查询分面扩展的方法能明显改善查询分面的挖掘效果. 这种方法一方面能够大幅度增加查询分面中词项的数目，提高分面词项的召回率；同时有助于发现和检测出原始查询分面中的部分噪声词项，带来分面精度的小幅提升.

本文采用的大规模词项图包含了词项与词项之间的关系以及词项与类别之间的关系，可以认为是一种简单的知识库. 如何利用知识库来生成高质量的分面是一个值得深入探讨的问题. 在后续工作中，将对这一问题进行进一步研究.

参 考 文 献

[1] China Internet Network Information Center. The 35th Chinese Internet Development Annual Report, 2015(in Chinese)
(中国互联网络信息中心. 第 35 次中国互联网络发展状况统计报告, 2015)

[2] Dou Zhi-Cheng, Hu Sha, Luo Yu-Long, et al. Finding dimensions for queries//Proceedings of the CIKM 2011. New York, USA, 2011: 1311-1320

[3] Kong Wei-Ze, Allan J. Extracting query facets from search results//Proceedings of the SIGIR 2013. New York, USA, 2013: 93-102

[4] Kong Wei-Ze, Allan J. Extending faceted search to the General Web//Proceedings of the CIKM 2014. New York, USA, 2014: 839-848

[5] Wang R C, Cohen W W. Language-independent set expansion of named entities using the web//Proceedings of the ICDM 2007. Omaha, USA, 2007: 342-350

[6] Wang R C, Cohen W W. Iterative set expansion of named entities using the web//Proceedings of the ICDM 2008. Pisa, Italy, 2008: 1091-1096

[7] Allan J. Relevance feedback with too much data//Proceedings of the SIGIR 1995. Seattle, USA, 1995: 337-343

[8] White R W, Bilenko M, Cucerzan S. Studying the use of popular destinations to enhance web search interaction//Proceedings of the SIGIR 2007. Singapore, 2007: 159-166

[9] Xu J, Croft W B. Query expansion using local and global document analysis//Proceedings of the SIGIR 1996. Zurich, Switzerland, 1996: 4-11

[10] Zhang Z, Nasraoui O. Mining search engine query logs for query recommendation//Proceedings of the WWW 2006. Edinburgh, UK, 2006: 1039-1040

[11] Herdagdelen M, Ciaramita M, Mahler D, et al. Generalized syntactic and semantic models of query reformulation//Proceedings of the SIGIR 2010. Raleigh, USA, 2010: 283-290

[12] Zhang Hui-Bin, Zhu Ming-Jie, Shi Shu-Ming, Wen Ji-Rong. Employing topic models for pattern-based semantic class discovery//Proceedings of the Annual Meeting of the Association for Computational Linguistics(ACL'09). Singapore, 2009: 459-467

[13] Shi Shu-Ming, Zhang Hui-Bin, Yuan Xiao-Jie, et al. Corpus-based semantic class mining: Distributional vs. pattern-based approaches//Proceedings of the 23rd International Conference on Computational Linguistics(COLING'10). Beijing, China, 2010: 993-1001

[14] Jarvelin K, Kekalainen J. Cumulated gain-based evaluation of IR techniques. ACM Transactions on Information Systems, 2002, 20(4): 422-446

DOU Zhi-Cheng, born in 1980, Ph.D., associate professor. His research interests include information retrieval, data mining, and big data.

JIANG Zheng-Bao, born in 1993, M. S. candidate. His research interests include data mining and Web search.

LI Jin-Xiu, born in 1991, M. S. candidate. Her research interest is data mining.

ZHANG Yi-Chun, born in 1978, Ph. D. His research interests include related information processing, and basic theory on art and technology application.

WEN Ji-Rong, born in 1972, Ph. D., professor. His main research interests include information retrieval, database, big data, and data mining.

Background

The problem discussed in this paper is related to the fields of information retrieval and Web search. Search has been a powerful tool for people to seek information from the Web. With the help of search engines, users can quickly find

web pages related to what they want. By issuing a simple keyword query to a search engine, a user receives a list of search results. For some queries, the user can quickly get what they want by just clicking the first or a few results. But for some other queries, especially the queries the user used for surveying or analyzing a topic, she has to spend much time on clicking, viewing and summarizing these web pages by herself. This situation gets worse especially when the network speed is slow or the query contains many different aspects.

An effective method for solving this problem is mining query facets, where each facet is a significant list of information nuggets that explain an underlying aspect of the query. Several algorithms for mining query facets have been developed, including QDMiner, QF-I, and QF-J. A common problem of the existing algorithms is that they mainly rely on the top search results from search engines. More specifically, facets are generated by extracting lists contained in the search results. The coverage of facets mined using this kind of methods might be limited, because some useful facet items might not appear in a list within the search results used.

In order to solve this problem, we propose mining query facets by using a term graph mined from a large number of web pages in this paper. By leverage this term graph, we are able to discovery more facet terms that are not covered by the content of the query's search results or a list within the results, hence improve the quality of mined facets.

The authors of the paper first proposed the concept of query facet (or query dimension), and introduce QDMiner, the first query facet mining system in CIKM 2011 (Please refer to Ref. [2]).

This work was supported by the National Natural Science Foundation of China (Grant No. 61502501) and the National Key Basic Research Program (973 Program) of China under Grant No. 2014CB340403.