DB&IR 系统研究综述

林子雨[1] 左思强[1] 赖永炫[2] 张东站[1]
[1](厦门大学 计算机科学系, 厦门 361005)
[2](厦门大学 软件学院, 厦门 361005)

(ziyulin@xmu.edu.cn)

摘要 介绍了从DB、IR到DB&IR的发展历程；阐述了DB&IR系统的设计考虑因素；讨论了DB&IR系统的体系架构设计问题；介绍了DB&IR系统所采用的基于关键词的查询技术；最后总结并展望未来的研究方向。

关键词 关键词查询；数据库；信息检索

中图法分类号 TP301

DB&IR System: A Survey

Lin Ziyu[1], Zuo Siqiang[1], Lai Yongxuan[2], Zhang Dongzhan[1]
[1](Department of Computer Science, Xiamen University, Xiamen 361005, China)
[2](School of Software, Xiamen University, Xiamen 361005, China)

Abstract The development process from DB, IR to DB&IR is presented first, followed by the detailed description of the consideration factors for the design and architecture of DB&IR system. Then keyword search in DB&IR system is discussed. Finally, some future trends in this area are discussed.

Keywords keyword search; database; information retrieval

互联网信息检索（Information Retrieval，简称 IR）工具，比如 Google、Yahoo 和百度等，为人们的日常生产和生活中的信息获取提供了强大的技术支持。用户只需要输入关键词，就可以立刻得到大量相关的搜索结果，即方便易用，又快捷丰富。数据库（Database，简称 DB）为结构化数据存储和管理提供了完善的功能，可以让用户通过结构化查询语言（比如 SQL）发起精确的查询。在很长一段时间内，IR 和 DB 是彼此独立的两个发展领域。随着企业应用的发展，用户需要对结构化数据和文档实现无缝地集成和访问，这就产生了一个很自然的需求，即让关系数据库支持高效的关键词查询，并最终实现有效融合 DB 和 IR 功能的 DB&IR 系统。目前，这个方面的研究已经成为数据库领域比较热门的研究话题[1]。

本文第 1 节讨论从 DB、IR 到 DB&IR 的发展历程；第 2 节阐述了 DB&IR 系统的设计考虑因素；第 3 节讨论 DB&IR 系统的体系架构设计问题；第 4 节讨论了 DB&IR 系统所采用的基于关键词的查询技术；最后，在第 5 节做总结并展望未来的研究方向。

1. 从 DB、IR 到 DB&IR

DB 和 IR 作为两个独立的研究领域，都经历了很长的发展时期。三十多年以前，DB 是由工资单和存货清单管理等应用驱动发展起来的，而 IR 则早期主要应用于出版物和专利管理。二者针对各自的应用领域建立了成熟的理论体系，彼此的研究也各有侧重。

收稿日期：2010-06-20 Date 六号
基金项目：国家自然科学基金 No. 50604012

DB 关注结构化数据的存储和查询，主要研究数据的一致性，以及精确查询的高效性，在 DB 系统中，查询被视为一个基于逻辑谓词的匹配任务，DB 系统希望用户提出精确的查询，然后系统尽快给出精确的结果，这时的用户实际上扮演了技术人员的角色，需要掌握类似 SQL 的结构化查询语言。IR 则面向文档等非结构化数据，主要强调基于统计学知识的排序模型以及用户满意度，IR 系统把查询处理看成是基于统计模型的排序任务，并认为查询是近似的，只要尽力满足用户需求即可，不需要给出精确的结果，通常采用一个交互过程引导用户获得最终结果，对于 IR 系统而言，用户是一个非技术人员，具有认知能力和知识局限性。

在十多年以前，研究 DB 和 IR 的两个群体都开始意识到了 DB 和 IR 系统融合的必要性，因为，越来越多的企业应用开始同时需要结构化数据和文档。在此后的时间里，不断有新的研究成果出现。在过去的几年里，DB 研究群体已经做了许多研究工作，他们已经把近似查询和 top-*k* 查询能力增加到了数据库当中，而 IR 研究群体则更加关注半结构化数据（比如 XML 数据）和结构化数据（关系数据库）。

但是，目前还没有出现一个真正实现 DB 和 IR 的有效集成，并提供完备的 DB 和 IR 的相关功能的 DB&IR 系统。一个完善的 DB&IR 系统是一个“结构化数据+文档”的系统，一个平台和工具的集合，这些平台和工具以面向特定应用的方式进行即席组合。当使用这样一个系统时，应用开发者面临的复杂性是空前的。DB 和 IR 系统集成已经被列为数据库研究领域中一个比较具有挑战性的问题[1]。

2. 系统设计考虑因素

DB&IR 系统的设计是一项系统工程，通常需要考虑多方面的因素，其中的几个关键因素如下：

（1）灵活的评分和排序：对于一个真正功能强大的 DB&IR 系统来说，灵活可定制的评分和排序功能是核心，评分函数应该可以针对任何结构化和非结构化数据。

（2）强大的搜索引擎：引擎必须具备三个基本的属性。第一，不能丢失相关的结果，第二，必须是高效的，第三，生成的答案的顺序必须和用户期望的顺序高度接近。判断引擎高效性的可以采用的参考标准是，引擎是否能够在多项式时间延迟[2]内枚举答案。

（3）数据模型：在设计上应该体现结构化数据独立性，保证用户可以对结构化和非结构化数据发起比较复杂的查询和关键词查询[3]。

（4）灵活而强大的查询语言：系统可以允许返回任意的结果，以及使用任意的评分函数。同时，系统应该支持各种类型的关键词查询形式，比如短语查询、单词查询、布尔查询、近似查询和模式匹配等。

（5）可优化性：DB&IR 系统中的查询应该可以利用一些负载和数据特性进行优化处理。尤其是，评分和排序操作需要放置在查询处理计划中进行考虑。而且，当前面的几个查询结果就可以满足要求时，查询处理器就应该及时停止查询处理工作[1]。

3. 体系架构

目前还没有一个统一的系统来管理结构化和非结构化数据，以及处理精确查询和排序查询。绝大多数系统都只是简单地把 DB 和 IR 引擎“粘合”在一起，而没有对二者做根本性的改进。DB&IR 系统的体系架构设计，主要有以下几种可能的方法：

- 在 DB 中增加 IR 功能
- 在 IR 中增加 DB 功能
- 中间件
- 全新的体系架构
- 其他

3.1 在 DB 中增加 IR 功能

IR 部分的功能构建在 DB 的功能完善的 SQL 引擎基础之上，从而使得 DB 系统具有 IR 的功能。但是，一些研究人员在 SIGMOD2005 的一个 DB&IR 专题研讨会上发表观点，认为这种方法不可行[3]。其中，Thomas Rolleke 认为，在经典的 SQL 技术之上搭建 IR 应用，可能无法满足 IR 应用的需求和可扩展性要求。Jayavel 认为，对 RDBMS 进行扩展，会违反当前数据库系统的许多假设，比如，对于一些属性的数据类型设计，传统的数据库都会采用一些默认的类型，但是，在 DB&IR 系统中应该采用文本类型还是结构化化数据类型呢？除此以外，数据库中的操作符都具有良好定义的精确的语义，而在 IR 当中，就连查询结果都没有经过很好的定义。而且，当 IR 需要返回 top-*k* 个排序结果时，不得不从 SQL 中抽取所有完整的结果，这显然不符合 IR 系统的特性。如果才能填平这二者之间的“鸿沟”呢？目前缺少有效的方法。

3.2 在 IR 中增加 DB 功能

这种方法和上面方法类似，也提供了一个对 DB&IR 功能进行有限集成的系统，这对于某些应用还是不错的解决方案，这些应用通常主要关注一种数据类型，即非结构化数据，同时，适量处理另一种数

据类型，即结构化数据。但是，Jayavel[3]认为这种方法同样不可行，因为，IR 系统很少提供对结构化数据的支持，评分函数也无法把数据的结构考虑在内。

3.3 中间件

集成 IR 和 DB 的系统，是由中间件层提供的，这个层位于 IR 系统和 SQL 引擎这两者之上。但是，这种方法把 DB 和 IR 系统都当作“黑盒子”来处理，没有真正实现 DB 和 IR 系统底层设计的融合，因此，无法提供强大的功能。

3.4 全新的体系架构

它可以取代今天的 DB 和 IR 系统，在设计方面，无论是数据模型、评分函数，还是查询语言，这种全新的系统都可以全面支持针对结构化数据和非结构化数据的统一处理。Thomas Rolleke[3]认为，在新的体系架构中，需要对关系代数核心进行更改，从而满足 IR 应用的需求。但是，这种系统是一种理想的方案，它的设计难度也是最大的。

3.5 其他方案

此外，文献[1]还给出了另外几种可能的设计方案，比如（1）通过 ADT（Abstract Data Type）的 IR：使用一个单独的引擎把 IR 功能集成到 SQL 引擎中，这个引擎可以采用特定的机制被关系数据库调用，比如用户自定义函数或 ADT；（2）RISC（Reduced Instruction Set Computing）：IR 查询服务层位于关系存储层之上。关系存储层是一个核心系统，它与 System R 的 RSS，Exodus+Shore，Berkeley DB 等一样，具备受限制的 RISC 类型的功能和简单的 API。DB&IR 系统会形成位于存储层之上的单独层，应用就需要在 DB&IR 系统 API 基础之上构建。作者认为，RISC 方法提供了最小的复杂性。和其他方法相比较，RISC 方法具有两个方面的优点：（1）可定制性与编程复杂性；（2）高效性。

4. 关键词查询

这部分内容介绍 DB&IR 系统中的关键词查询技术，主要介绍了基于数据图的方法和基于模式图的方法，并比较了二者的优缺点。

4.1 概述

DB&IR系统采用基于关键词的查询方法，即用户只需要给出查询关键词，系统就可以返回相应的答案，不需要用户掌握复杂的结构化查询语言（比如 SQL）和数据库模式知识。和IR领域的关键词查询相比，DB领域的关键词查询显得更加复杂。在IR领域，关键词查询所面向的对象是文档的集合，查询结果通常只包含单个文档。而对于DB中的关键词查询而言，关键词查询所面向的对象则是数据库，查询结果并不是单个元组，而是包含多个元组的元组连接树[4]，这些多个元组作为一个整体包含了查询中的全部关键词。

4.2 核心思想

实际上，如果把数据库看成一个图G，其中，数据库元组是节点，元组之间的主外键关联是边，那么，一个关键词查询的结果，就是一棵图G关于关键词K的简化子树T，即T包含了K，但是，不会再有T的子树包含K。由此，关键词查询问题就可以表示为一个简化子树枚举问题，这就是几乎所有DB&IR系统（比如DBXplorer[4]、BANKS[5]和DISCOVER[6]）的关键词查询方法的核心思想。

4.3 两种简化子树生成方法

到目前为止，研究人员已经提出了大量的简化子树生成方法[7-12]。概括起来，这些方法可以归入以下两个类别，即基于数据图的方法和基于模式图的方法，具体如下：

（1）基于数据图的方法：对于基于数据图的方法而言，它把数据库表示成图的形式，并且直接对图进行处理，从中枚举简化子树。由于 XML 数据和关系型数据等都可以建模成数据图，因此，该方法可以处理针对这些数据的关键词查询。同时，研究人员发现[8]，简化子树的枚举和 Steiner 树问题之间存在着对应关系，于是，简化子树枚举问题又被转化成等价的 Steiner 树问题。Steiner 树问题是一个 NP-hard 问题，已经被研究了很多年，诞生了大量的研究结果[13]，因此，这些研究可以被用来开发枚举简化子树的高效算法。采用基于数据图的方法的相关研究主要包括 BANKS[5]、BANKS-II[7]、BLINKS[14]以及文献[8-10,15]等。

（2）基于模式图的方法：基于模式图的方法则只能用于关系型数据的关键词查询，它会利用数据库模式创建连接表达式，然后在DBMS上执行连接表达式对应的SQL语句得到结果。基于模式图的方法的相关研究主要包括DBXplorer[4]、DISCOVER[6]以及文献[9,11]等。

4.4 两种方法比较

基于数据图的方法所采用的数据图模型比较直观，并且可以利用一些针对 Steiner 树问题的现成的

解决方案，但是，该方法也有缺点。首先，一些基于数据图的方法[5,7,10]可能丢失高度相关的结果；其次，存在可扩展性问题，因为，该方法必须对整个数据图进行遍历，而数据图的大小通常比模式图大好几个数量级，另外，为了找到了包含关键词的元组之间的连接关系，算法要执行大量的元组连接操作[12]。

基于模式图的方法可以很好地利用了数据库模式，加快了算法执行速度。但是，这种方法同样存在不足之处。对于基于模式图的方法而言，绝大多数研究，比如 DBExplorer、DISCOVER 和 DISCOVER-II，都会生成所有可能包含查询关键词的连接表达式。如果数据库模式比较小并且结构简单，那么，生成所有连接表达式的过程就会非常迅速。但是，通常来说，对于实际应用中的数据库来说，许多连接表达式可能最终只生成空结果[15,16]，这就降低了整个查询过程的效率。

5. 总结与研究展望

DB&IR 系统有效融合 DB 和 IR 两个领域的相关技术，可以支持针对结构化数据和文档的统一形式的关键词查询。本文介绍了 DB 和 IR 的发展历程，讨论了 DB&IR 系统在设计时需要重点考虑的几个方面的关键因素以及几种系统体系架构设计方法。关键词查询是 DB&IR 系统需要解决的一个核心问题，本文讨论了数据库中的关键词查询的特点、核心思想和解决方案。

纵观现有的 DB&IR 系统的相关研究，我们认为，以下几个方面是该问题未来的研究方向：

（1）**性能评价机制**：目前已经有大量与 DB&IR 系统相关的研究，但是，却缺少统一的性能评价机制，许多研究大都采用了不同的数据集和不同的性能指标，并从不同角度来衡量算法或系统的综合性能。缺少一个统一的性能评判机制，就使得一些性能结果对比测试很难具有普遍说服力。因此，Chen 等人[17]在 2009 年的 SIGMOD 大会的专题报告上，指出了建立性能评价机制的必要性，从而有助于今后的系统设计，主要内容是建立一套测试基准（benchmark）和形式化评估体系。

（2）**成熟的 DB&IR 系统**：DB&IR 系统具有很大的市场需求，DB 和 IR 的融合显得尤为迫切。从目前已有的研究来看，绝大多数系统（比如 BANKS、DBExplorer、DISCOVER、ObjectRank 和 NUITS 等）都无法满足本文第 3 节介绍的关于 DB&IR 系统应该具备的架构和功能要求，比如，有的系统缺少灵活的评分和排序机制、可能丢失高度相关的结果、不具备相关的高级查询优化机制、没有提供强大的查询语言等等。不同系统都只是在某一方面取得了显著的成果，比如，DBExplorer 和 DISCOVER 可以很好利用关系数据库引擎的功能，BANKS 具备灵活的评分和排序机制，NUITS 具有较好的结果呈现机制；但是，它们都无法成为满足商业应用的成熟系统。DB 和 IR 的融合是一个复杂的系统工程，需要等待多方面的相关技术的不断成熟和完善。因此，我们可以认为，鉴于这项工作的艰巨性和复杂性，它会是今后 DB 和 IR 研究群体在很长一段时期内的工作重点。

参 考 文 献

[1] Chaudhuri S, Ramakrishnan R, Weikum G. Integrating DB and IR technologies: what is the sound of one hand clapping? [C] // Proceedings of the 2nd biennial conference on innovative data systems. New York: ACM, 2005: 1-12

[2] Johnson D, Papadimitriou C, Yannakakis M. On generating all maximal independent sets [J]. Information Processing Letters, 1988, 27(3): 119-123

[3] Amer-Yahia S, Case P, Rölleke T, et al. Report on the DB/IR panel at SIGMOD 2005 [J]. SIGMOD Record , 2005, 34(4):71-74

[4] Agrawal S, Chaudhuri S, Das G. DBXplorer: a system for keyword-based search over relational databases [C] // Proceedings of the 18th International Conference on Data Engineering. Piscataway, NJ: IEEE, 2002: 5-16

[5] Bhalotia G, Hulgeri A, Nakhe C, et al. Keyword searching and browsing in databases using BANKS [C] //. Proceedings of the 18th International Conference on Data Engineering. Piscataway, NJ: IEEE, 2002: 431-440

[6] Hristidis V, Papakonstantinou Y. DISCOVER: keyword search in relational databases [C] // Proceedings of 28th International Conference on Very Large Data Bases. San Fransisco: Morgan Kaufmann, 2002: 670-681

[7]Varun Kacholia, Shashank Pandit, Soumen Chakrabarti, S. Sudarshan, Rushi Desai, Hrishikesh Karambelkar: Bidirectional Expansion For Keyword Search on Graph Databases. VLDB 2005:505-516.

[8] Kimelfeld B, Sagiv Y. Efficient engines for keyword proximity search [C] // Proceedings of the Eight International Workshop on the Web and Databases. New York: ACM, 2005:67-72

[9] Liu Fang, Yu C, Meng Weiyi, Chowdhury A. Effective keyword search in relational databases [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 2006: 563-574

[10] Ding Bolin, Yu J, Wang Shan, et al. Finding top-k min-cost

connected trees in databases [C] // Proceedings of the 23th International Conference on Data Engineering. Piscataway, NJ: IEEE, 2007: 836-845

[11] Luo Yi, Lin Xuemin, Wang Wei, Zhou Xiaofang. Spark: top-k keyword query in relational databases [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 2007: 115-126

[12]Dalvi B, Kshirsagar M, Sudarshan S. Keyword search on external memory data graphs [J]. PVLDB, 2008, 1(1):1189-1204

[13] Robins G, Zelikovsky A. Improved Steiner tree approximation in graphs [C] // Proceedings of the ACM-SIAM Symposium on Discrete Algorithms. New York: ACM, 2000: 770-779

[14] He Hao, Wang Haixun, Yang Jun, et al. BLINKS: ranked keyword searches on graphs [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 2007: 305-316

[15] Kimelfeld B, Sagiv Y. Efficiently enumerating results of keyword search over data graphs [J]. Information System, 2008, 33(4-5): 335-359

[16] Golenberg K, Kimelfeld B, Sagiv Y. Keyword proximity search in complex data graphs [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 2008: 927-940

[17]Chen Yi, Wang Wei, Liu Ziyang, Lin Xuemin: Keyword search on structured and semi-structured data [C] // Proceedings of the ACM SIGMOD International Conference on Management of Data. New York: ACM, 2009: 1005-1010

林子雨，男，1978 年生，助理教授，主要研究方向为数据仓库、OLAP 和数据挖掘。

左思强，男，1984 年生，硕士研究生，主要研究方向为数据仓库、OLAP 和数据挖掘。

赖永炫，男，1981 年生，助理教授，主要研究方向为传感器数据库、数据流、数据仓库和数据挖掘。

张东站，男，1974 年生，副教授，主要研究方向为数据仓库、OLAP 和数据挖掘。