第43卷 第6A期
2016年6月

计 算 机 科 学
Computer Science

Vol. 43 No. 6A
June 2016

基于索引视图实现动态数据仓库的实时数据加载

武 彤 谭光炜
(贵州大学计算机科学与技术学院 贵阳 550025)

摘 要 随着数据仓库技术的不断普及,基于数据仓库技术的决策支持系统在企业得到了广泛应用,同时出现了动态数据仓库。随着动态数据仓库在决策支持领域扮演的角色越来越重要,企业利用决策支持系统从辅助进行战略决策开始向战术决策转变。而要进行战术性的分析,数据仓库中必须有动态变化的最新数据,实现数据仓库的"动态"特性的关键又在于实现"动态数据获取",即实现实时数据捕获加载。提出了基于索引视图实现动态数据仓库的实时数据加载,并通过实验验证了其可行性。该方法对进一步深入研究实时数据捕获技术有一定的借鉴作用。

关键词 动态数据仓库,数据仓库,实时数据,数据加载
中图法分类号 TP309.2 文献标识码 A

Real-time Data Loading of Dynamic Data Warehouse Using Index View Set

WU Tong TAN Guang-wei
(College of Computer Science & Technology, Guizhou University, Guiyang 550025, China)

Abstract With the vast employment of data warehouse technology, decision making supporting system using data warehouse technology is commonly used among corporations. At the same time, dynamic data warehouse appeares. As dynamic data warehouse becomes increasingly important in the decision making supporting field, corporations tend to more frequently employ decision making supporting system to assist tactical decision making, migrating from strategical decision making. However, the dynamically updated data in the data warehouse is the prerequisite to carry out tactical analysis. The key to realize the dynamic features of data warehouse is to obtain dynamic updated data, which is real-time data loading. This article proposed to use index view set to realize real-time data loading in the dynamic data warehouse and demonstrated the feasibility of the approach. It also shields light to further research on real-time data loading.

Keywords Dynamic data warehouse, Data warehouse, Real-time data, Data loading

数据仓库发展之初的主要作用是为企业内部的某些部门提供一些固定的报表,这一阶段通常被称为"报表"阶段。当企业的关注点从"发生了什么"转向"为什么会发生"后,数据仓库进入了"分析"阶段。这一阶段,决策者开始对数据进行分析,实质上是了解报表数据的真实涵义,这就需要更详细地对数据进行多角度分析。接下来就要将业务信息用于预测。数据仓库随之进入"预测"阶段,即数据挖掘阶段。数据挖掘能够预知企业即将发生的动向,帮助管理者积极地管理和实施企业战略。数据仓库演进的第4阶段是动态数据仓库。动态数据仓库技术在企业环境成熟应用后,将引领企业"动态性"阶段。伴随动态数据仓库在决策支持领域扮演的角色越来越重要,企业实现决策自动化的积极性也在不断提高。为了寻求决策的有效性和连续性,企业会趋向于采取自动决策方式。

"动态数据仓库"是一种创新理念,但其技术基础和架构思想还是来自传统数据仓库[1]。关键的区别是动态数据仓库增加了"动态"特性,即如何实现数据仓库的"动态"特性,这也是部署"动态数据仓库"系统的关键所在。而关键的关键又在于如何实现动态数据仓库的动态数据获取。也就是说,动态数据仓库要进行战术性的分析和事件的检测与处理,就必须有最新的数据才能保证决策和事件处理的有效性,因此数据必须实时或接近实时地加载到动态数据仓库中。要实现动态数据仓库的实时数据加载,就必须实时有效地捕获到业务系统中的变化数据,然后进行加载。

本文研究了基于索引视图实现动态数据仓库的实时数据加载方案,并通过实验验证了方案的可行性。

1 常用的变更数据捕获技术

在动态数据仓库中,不只综合表中的汇总数据用于战略决策分析,细节表中的数据也需要响应实时的战术决策。为了满足实时性的决策需求,数据源中的数据发生变化时需要实时地反映到细节表中,这样就产生了对变更数据捕获技术的需求。与传统数据仓库集成数据的ETL方式相比,变更数据捕获技术强调得到变更数据的即时性[2],即数据源中数据发生变化时便即刻捕捉到此变更的数据。第二个关键的不同点在于使用ETL方式集成数据时,作为数据源的操作型系统只能停止工作,不能响应外部操作请求,而变更数据捕获技术获取数据源中变更的数据时,不应该影响源系统的正常运行。

本文受动态数据仓库的数据加载技术研究(黔科合J字[2013]2115号)资助。
武 彤(1964—),女,硕士,教授,CCF会员,主要研究方向为数据仓库;谭光炜(1990—),男,硕士生,主要研究方向为数据库技术。

目前常用的变更数据捕获技术有下面几种。

1.1 基于快照差分的方法

以捕获数据前后数据源的快照作为根据，得到一个捕获周期内的变更数据。其总体流程是设定一定的周期，每隔一个周期扫描一次数据源表的快照，将其与上个周期扫描的历史快照做差分计算，得到该周期内数据源发生变化的数据。计算完成之后，将此次的数据源快照暂存下来，作为下一次快照差分计算的历史快照[3]。

基于快照差分的方法适用于任何数据格式的数据源，在技术上较易实现，并且此方案在实现上具有一定的通用性，实现方法的可移植性较高，不过这种方案需要将变更前后的所有源数据加载进内存，并进行差分计算，内存和时间消耗均比较大，实时性也不能保证。该思路适用于对实时性要求不是很高的准实时数据的数据捕获[4]。

1.2 基于触发器的方法

对于数据源是关系数据库表的情况，可以在作为数据源的数据库表上设置触发器，在触发器的触发动作体中编写存储过程，使得每当数据源表出现变更时，由触发器捕获数据变化情况，将数据变更写入一个记录表中，再后续地将变更数据同步到数据仓库的表中[5]。基于触发器的方法要求数据源所在的数据库管理系统与数据仓库所在的数据库管理系统必须一致。

触发器机制基于事件驱动捕获变更数据，可以保证捕获的实时性，且触发动作体中实现数据捕获技术难度不大，不过触发器作为一种数据库系统约束机制，其频繁的触发会给数据库系统带来较大的负担。触发器适用于对实时性要求较高且建立触发机制的表数据变更频率较低的数据表的数据捕获。

1.3 基于分析日志的方法

各主流数据库管理系统都提供事务日志文件记录表中数据的操作情况，可以基于事件驱动机制，实时地读取事务日志记录并根据其数据组织格式对日志记录进行分析，从中得到作为数据源的数据库表的操作变更记录[6]。

基于事务日志的方案同样基于事件驱动机制触发读取、分析事务日志的操作，可以保证数据捕获的实时性，且对日志的访问和分析都在操作系统层上进行，不会给源系统带来过重的负担。不过事务日志的数据格式复杂，对它的分析实现有一定难度，且要求数据源提供事务日志及其访问接口。这种方案适用于对数据捕获的实时性要求较高、数据源是数据库系统中的表且提供事务日志访问接口的场景。

由以上分析可知，目前常用的变更数据捕获方法各有其优缺点，在具体使用时必须根据需求选择使用。

2 索引视图的基本概念

数据库与数据仓库中都存在大量视图，普通的视图在数据库管理系统中只在数据字典中保存其定义，实际数据不保存，在对视图进行查询和操作时，实际是转化为对建立视图的基表的操作，因此可称之为“虚视图”。与之相对，有物化视图的概念(SQL Sever 中称为索引视图，Oracle 中称为物化视图)，这类视图是一种数据被实体化的视图，定义这类视图时，被选取的数据会实际保存在数据库管理系统中，是一个真正存在的关系。物化视图的出现提高了数据访问速度，对于高频率的复杂查询，索引视图的优点更加明显。

相对于不实际保存数据的普通视图，形成了实体化视图的概念，在各大主流厂商的数据库管理系统中，都有建立实体化视图的对应方法。本文研究平台的数据仓库基于 SQL Server 2008，因此以 SQL Server 2008 为例说明实体化视图的概念。在 SQL Server 中，实体化视图被称为索引视图，顾名思义，索引视图就是含有索引的视图，在 SQL Server 中，当在普通视图上创建一个唯一聚簇索引(uniqueclusterd index)时，这个视图的结果集将会被实体化并保存在数据库管理系统的物理存储设备中，即 SQL Server 将实体化这个视图，使之成为一个索引视图。创建聚集索引之后，视图被实体化，在此视图上还可以继续创建其他非聚集索引[7]。

创建索引视图的一般命令：

CREATE VIEW [view-name] WITH SCHEMABINDING AS Select [column-name] From [table-name] Where [条件表达式] Group BY [column-name]

作为实体化的视图，索引视图的结果集会被具体化并保存在数据库管理系统的物理存储中。普通视图在查询时会在每次查询时都进行一次查询涉及到的聚集和连接运算。与之相对，索引视图会：1)预先计算聚合并将其保存在索引中，从而在查询执行时最小化高成本的计算；2)预先联接各个表并保存最终获得的数据集；3)保存联接或聚合的组合[8]。因此，在查询时节省了计算的成本，改善了查询效率。

在动态数据仓库的数据加载过程中，需要对作为数据源的数据库进行查询以抽取变更数据，由于数据仓库的表对应数据源数据库中多个不同的表，而数据仓库表中往往又会有一些新的导出字段，因此实时地获取变更数据会导致频繁的数据连接和聚合操作，给作为数据源的数据库系统带来很大的压力。在这样的情况下，可以尝试将索引视图引入动态数据仓库的实时数据加载方案，以期减小实时获取变更数据的过程对数据源数据库系统的影响。

3 基于索引视图进行变更数据捕获

3.1 研究平台简介

3.1.1 研究平台的总体结构

本文的研究基于某公司的电视机生产线产品质量控制系统，该产品质量控制系统由生产线数据采集及 OLTP 系统和生产线质量决策分析系统两个子系统组成。数据采集及 OLTP 系统的主要用途是从生产线现场的数个数据采集点采集电视机生产中的产品质量信息，并将其存入事务处理系统对应的数据库 PQACS 的电视机质量信息记录表 faultstatistics 中，以便进行联机事务处理；生产线质量决策分析系统的主要功能包括 OLAP 及数据挖掘，分析处理的数据基于数据仓库 HXDW，决策分析系统除了可以用于 OLAP 等战略分析外，还可以在此基础上根据具体需求扩充其应用范围，提供对实时决策的支持。整个系统的结构如图 1 所示。

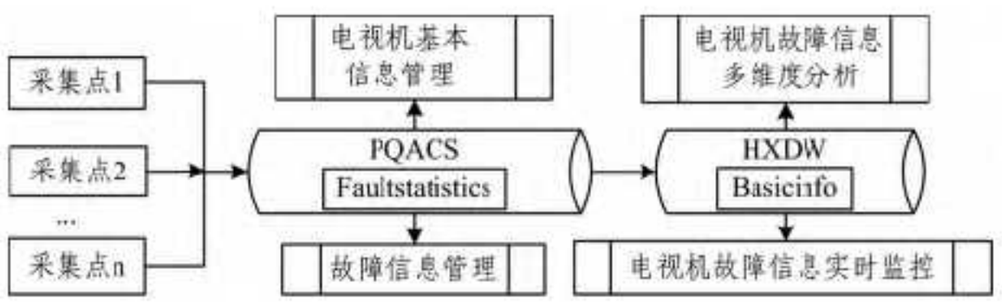

图 1 生产线质量控制系统的总体结构

图 1 中，在生产线上有 8 个故障信息采集点，以扫码的方式采集故障信息，故障信息数据会存入 OLTP 子系统对应的数据库中(PQACS)，在数据库中有一张专门的故障信息统计表(faultstatistics)，该表数据主要用于联机事务处理。决策分析系统的数据对应电视机故障信息数据仓库(HXDW)，该数据仓库的数据源为 OLTP 系统的数据库(PQACS)，从数据库中导入数据仓库的故障信息数据会首先导入数据仓库中的一张故障信息细节表中(basicinfo)，该细节表中的数据按照电视机的种类组织成 4 类数据，分别导入到第二级的 4 张细节表中。之后按故障现象、生产线、电视机种类等维度进行组织，从这些细节表中分别得到按天、按月、按季度、按年不同数据综合程度的综合表。传统的 OLAP 分析均基于综合表进行。

3.1.2 实时决策分析需求

基于数据仓库(HXDW)，用户提出了一系列实时查询分析的需求，如：当某一类型的电视机产品每天出现故障的数量超过给定阈值时，需要立即检查生产线上的生产情况；当由于同一故障原因出现的故障数量超过给定阈值时，需要立即检查原材料质量、线上生产工人状态等。

数据仓库中所有的故障统计信息都集中在细节表 basicinfo 中，这时就需要将 PQACS 数据库中的故障记录信息表 faultstatistics 中的新数据实时地加载进数据仓库中的故障统计表 basicinfo 中。根据用户方提出的要求，需要在 10 秒内基于最新的故障数目信息做出报警。因此这里的实时加载也是指在满足需求的前提下尽量缩短延时。

3.2 索引视图的设计

根据上面所述的实时决策分析需求，分别采用了前面介绍的几种常用的变更数据捕获方法进行了实验，发现在实时加载数据的过程中，需要周期性地查询前端数据源的变更数据表，每次查询还需要做大量连接其他电视机基本信息表的操作。为了减小数据加载过程中查询数据的代价，尝试引入索引视图，并通过实验验证此方法可进一步提高实时数据更新方案的性能。

在数据仓库的当前细节数据中，有些数据可以直接从前端数据源的某一个数据表中提取，而有些数据必须从前端数据源的几张数据表中提取。由此可知在每个周期的数据加载过程中，需要做大量多表连接操作，而在变更数据捕获及加载过程中除了考虑效率之外，还需考虑对前端数据源的影响，因此要减小数据加载过程中多表连接查询的代价。前文所述的索引视图是一种在数据库管理系统中不只存储定义还实际存储数据的视图，是一种实物化的视图，我们可以在前端数据源中以作为数据源的表为基表建立索引视图，将对表的操作转化为对索引视图的操作，有效地减小数据加载程序频繁查询数据库表带来的时间和资源消耗的代价，这是一种用空间换取时间的策略。下面给出两种建立索引视图的方法，并进行实验比较验证。

3.2.1 基于变更记录表建立索引视图

将前端数据库中的故障记录信息表 faultstatistics 中的变化数据记录在一张变更记录表 FaultStatistics_CT 中，以变更数据表 FaultStatistics_CT 作为基表之一，由此基表关联故障现象表(Faultapp)、故障原因表(Faultcause)、电视机信息表(TVinfo)建立索引视图。在这种思路下，只需查询一个索引视图和两张数据表即可完成一个周期内的实时数据加载工作。这种索引视图的建立思路如图 2 所示。

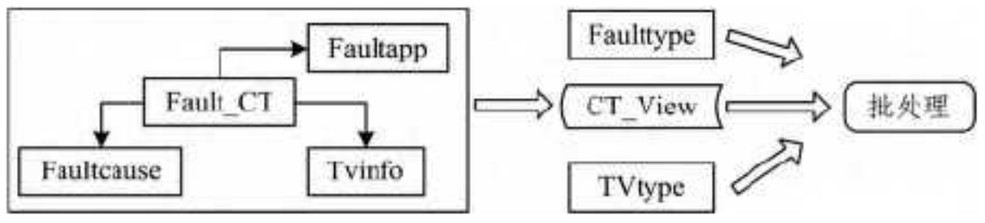

图 2 基于变更记录表建立索引视图

3.2.2 基于基本信息表建立索引视图

基于基本信息表建立索引视图是分别整合故障现象表(Faultapp)和故障类型表(Faulttype)以及电视机信息表(TVinfo)和电视机种类表(TVtype)建立两个索引视图，在这种思路下，一个周期内的实时数据加载工作转化为对两个索引视图和两张数据表的查询。这种索引视图的建立思路如图 3 所示。

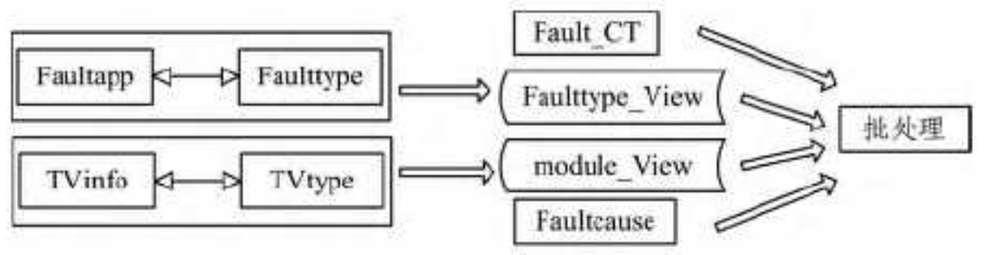

图 3 基于基本信息表建立索引视图

索引视图的运用可以带来诸多好处，但是也有其弊端，如果索引视图基于的基表数据变更频繁，则索引视图频繁的刷新操作也会消耗 DBMS 的大量资源，对前端数据库系统产生干扰，同时影响数据加载的实时性。第一种思路基于变更数据表建立索引视图，变更数据表是实时更新故障信息的表，其数据变化非常频繁，表中数据的变化会使索引视图也随之频繁更新，这是一笔不小的代价。第二种思路基于稳定、数据变化频率较小的基本信息表建立索引视图，不会产生过多索引视图刷新带来的资源和时间消耗，不过在每个数据加载周期内，需要在查询变更数据表的同时，额外查询两个索引视图的内容。这两种方案的具体效果在下文进行实验验证。

3.3 实验验证

采用同一个实验环境，以如上所述两种方案建立索引视图，比较数据捕获加载时间及加载过程对前端数据库系统的影响程度，来说明方案的优劣。

实验环境：生产线上的 8 个采集点以 0.5 次/s 的速度扫码，在前端 OLTP 系统一端录入故障信息，同时开启 SQL Server CDC 机制及数据加载程序(基于索引视图)。

图 4 所示是基于基本信息表建立索引视图的数据加载方案(dataload 方法)运行 5 次的数据加载时间。

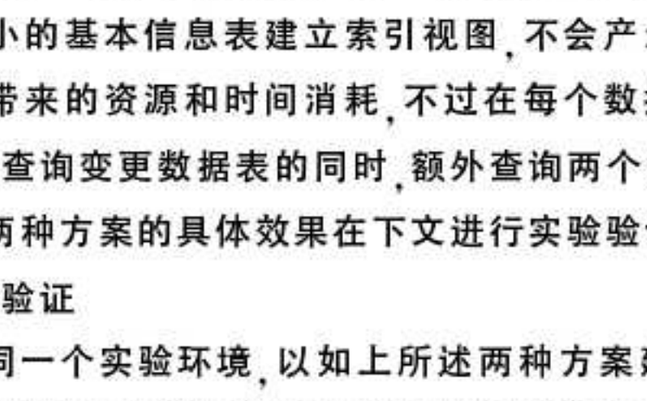

图 4 基于基本信息表建立视图 5 次 dataload 方法运行的时间

由图 4 可见，5 次 dataload 方法的运行时间分别约为 0.29s、0.27s、0.31s、0.28s、0.30s，平均时间 0.29s。同时在

前端数据库系统中进行 5 次 OLTP 操作，每次的响应时间分别为 2.585s、2.396s、2.624s、2.525s、2.479s，平均时间为 2.522s。

图 5 所示是基于变更数据表建立索引视图的数据加载方案(dataload 方法)运行 5 次的数据加载时间。

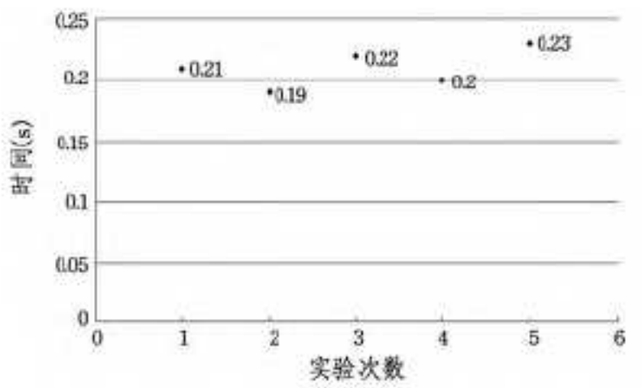

图 5 基于变更数据表建立视图 5 次 dataload 方法的执行时间

由图可见，5 次 dataload 方法的运行时间分别约为 0.21s、0.19s、0.22s、0.20s、0.23s，平均时间为 0.21s。同时在前端数据库系统中进行 5 次 OLTP 操作，每次的响应时间分别为 5.030s、5.203s、4.988s、5.146s、5.075s，平均时间为 5.088s。

通过实验数据可知，两种方法建立索引视图的数据加载方案对数据加载过程本身的效率影响不大。但基于变更数据表建立索引视图的思路，由于基表内容变化频繁，索引视图不断地刷新给前端 OLTP 系统带来了较大的压力，对事务操作的响应时间影响较大[9]。

通过以上分析可知，基于基本信息表建立索引视图的方案由于降低了一些连接和聚集操作的消耗，在数据加载的时间和对源系统的影响上都有所改进。而基于变更数据表建立索引视图的方案中每个周期的数据加载中只需要对一个索引视图进行查询，加载的实时性更好一些，不过由于变更数据表中数据变更频繁，维护索引视图时的数据刷新操作给源系统带来了很大的负担，事务处理系统的运行受到了较大的影响。

结束语 通过实验可以得出如下结论：采用基于基本信息表建立索引视图的方案不仅可以增强数据加载的实时性，还可以减小数据更新过程对源系统的影响。而采用基于变更数据表建立索引视图的方案可以更进一步增强数据加载的实时性，但是会给源系统带来更加沉重的负担。总之，索引视图是实物化的视图，实时的数据加载涉及频繁的连接和聚集操作，采用索引视图可以有效减小这些计算的代价。这一研究结论对动态数据仓库中实时数据加载方法的进一步研究有一定的借鉴作用，同时对特定的场合进行实时数据捕获时有一定的推广应用价值。

参 考 文 献

[1] iHaisten M. Real Time Data Warehouse: The Next Stage in Data Warehouse Evolution[J]. DM Review, 2003

[2] 徐富亮，周祖德. 变化数据捕获技术研究[J]. 武汉理工大学学报(信息与管理工程版)，2009，31(5)：740-743

[3] Ankorion I. Change data capture-efficient ETL for real-time BI [J]. DM Review magazine, 2005(1)

[4] 刘兆强. 基于快照差分的数据源更新检测方法研究及其实现[D]. 广州：暨南大学，2007：13-20

[5] 陆剑锋，张洁. 数据仓库数据更新的研究及基于 Oracle 数据库的开发与应用[J]. 计算机工程与应用，2004，40(26)：384-386

[6] 邹先霞，贾维嘉，潘久辉. 基于数据库日志的变化数据捕获研究[J]. 小型微型计算机系统，2012，3(3)：531-536

[7] 王珊，萨师煊. 数据库系统概论[M]. 北京：高等教育出版社，2009：89-91

[8] SQL Server 视图索引与索引视图指南[OL]. http://database.51cto.com/art/201007/212533.htm

[9] 谭光炜. 动态数据仓库实时数据的捕获及加载技术研究[D]. 贵阳：贵州大学，2015：50-63

(上接第 488 页)

量指标的有效合并。本文方法统筹兼顾了与用户查询词语义相似的词项和共现的词项，全部选取了与用户查询相似概率和共现概率都高的词项和与用户查询相似概率高或共现概率高的词项，有效地合并了语义扩展词集和统计扩展词集，极大地提升了搜索引擎的检索性能。

结束语 基于语义资料和局部分析混合式查询扩展可以同时提供具有语义相关和时效性的扩展结果，但如何有效混合两种相似度度量指标是尚未得到有效解决的问题，本文提出的基于 Copulas 框架的混合式查询扩展方法在统一框架内实现了不同类型相似度度量指标的合并。实验结果表明，该方法充分利用了语义及词语共现分析查询扩展方法的优点，有效地弥补了两者的不足，提高了搜索结果的查准率，比其它混合式查询扩展方法具有更优的搜索性能。

参 考 文 献

[1] Carpineto C, Romano G. A Survey of Automatic Query Expansion in Information Retrieval[J]. ACM Computing Surveys, 2012, 44(1): 1-50

[2] Selvaretnam B, Belkhatir M. Natural Language Technology and Query Expansion: Issues, state-of-the-art and Perspectives[J]. Journal of Intelligent Information Systems, 2012, 38(3): 709-740

[3] 李兴春. 信息检索技术中基于语义的扩展查询研究[J]. 重庆师范大学学报(自然科学版)，2013，30(4)：115-118

[4] Runkler T A, Bezdek J C. Automatic keyword extraction with relational clustering and Levenshtein distances[J]. Institute of Electrical and Electronics Engineers, 2002, 9(2): 636-640

[5] X Jin-xi, Croft B. Improving the effectiveness of information retrieval with local context analysis[J]. ACM Transactions on Information Systems, 2000, 18(1): 79-112

[6] 朱鲲鹏，魏芳. 基于用户日志挖掘的查询扩展方法[J]. 计算机应用与软件，2012，29(6)：113-115

[7] Pal D, Mitra M, Datta K. Improving Query Expansion Using WordNet[C]//CoRR. 2013: 1-18

[8] 王旭阳，萧波. 基于本体和局部上下文分析的查询扩展方法[J]. 计算机工程，2012，38(7)：57-59，69

[9] 吴秦，白玉昭，梁久祯. 一种基于语义词典的局部查询扩展方法[J]. 南京大学学报(自然科学)，2014，50(4)：526-533

[10] 欧阳柳波，谭容哲. 一种基于本体和用户日志的查询扩展方法[J]. 计算机工程与应用，2015，51(1)：151-155

[11] Sklar A. Fonctions de repartition an dimensions et leurs marges [J]. Publ. Inst. Statist. Univ. Paris, 1959, 8(1): 229-231

[12] Eickhoff C, de Vries A P, Collins-Thompson K. Copulas for Information Retrieval[C]//SIGIR'13. Dublin, Ireland, 2013: 663-672