大数据与关联数据:正在到来的数据技术革命*

刘　炜　夏翠娟　张春景
(上海图书馆上海科学技术情报研究所　上海 200031)

【摘要】当前越来越多的关联数据开始寻求突破关系数据库固有的局限,采用非关系型数据库(NoSQL)处理"大规模"的RDF数据。越来越多的大数据应用引入语义技术,通过语义链接,给大数据系统带来开放性和互操作性,并能提供基于"知识"的分析。通过介绍上述背景,区分"大"关联数据和"关联的"大数据两类不同的应用,对目前采用大数据技术发布关联数据的方法和路径进行梳理,同时对大数据领域应用关联数据技术的进展也做出介绍和点评,展望这两类数据技术在图情领域的发展前景。

【关键词】大数据　关联数据　语义网　数据技术　数字图书馆
【分类号】TP393

Big Data and Linked Data: The Emerging Data Technology for the Future of Librarianship

Liu Wei　Xia Cuijuan　Zhang Chunjing
(Institute of Scientific & Technical Information of Shanghai, Shanghai Library, Shanghai 200031, China)

【Abstract】Nowadays the ever growing linked data has broken through the restrictions between the triple structure and the relational model, and tend to use NoSQL approaches more often. More and more Big Data solutions provide semantic annotation and reasoning features. It brings machine readable semantics, rich meaningful linkage and knowledge analytics to the big data, and provide openness and interoperability to applications. The paper introduces the above background, makes difference between BIG Linked Data and LINKED Big Data systems, which the former implies the Big Data approach adopted by the Linked Data community, and the later vice versa. It also comments on the progress and benefit with the two cutting edge data technologies and gives outlooks on the future of the Big and Linked Data mashups.

【Keywords】Big Data　Linked Data　Semantic Web　Data technology　Digital library

作为语义万维网的实现方式,关联数据要求数据采用RDF三元组模型,这种模型更适宜采用图形式计算(Graph Model)而不是关系型数据库计算模式[1]。随着关联数据应用的数据量越来越大,越来越多的关联数据开始采用"大数据"解决方案,非关系型数据库(NoSQL)是大数据技术的典型代表,最适合存储RDF三元组的图形数据库(Graph DB)是4类NoSQL数据库之一[2]。目前,越来越多的大数据应用开始引入语义技术,使数据的描述更为规范且富含机器可理解的语义,丰富的语义链接使系统具有更好的开放性和互操作性,并使大数据的分析深入到"知识"层次,这就要求大数据技术能够高效支持RDF仓储(RDF Store),并能提供丰富的关联功能和简单的推理能力[3]。

收稿日期:2013-03-14
收修改稿日期:2013-04-07
* 本文系国家社会科学基金重大项目"基于语义的馆藏资源深度聚合与可视化展示研究"(项目编号:11&ZD152)和国家社会科学基金项目"关联数据的理论和应用研究"(项目编号:11BTQ041)的研究成果之一。

1　“大”关联数据与“关联的”大数据

关联数据是指以 URI 作为数据标识，以资源描述框架①(Resource Description Framework，RDF) 的三元组结构作为数据模型，并基于 HTTP 发布到互联网上的数据应用形式，是语义 Web 的一种简化实现，意图在目前以文档为基础的互联网之上构建“数据的 Web”[4]。“大数据”是指传统的数据库技术(通常指关系数据库系统) 无法很好地提供管理工具的海量、非结构化或半结构化数据集[5]。大数据技术能够高效地解决分布式环境下全网域(Web Scale) 的非结构化信息的管理和利用问题，而关联数据所带来的丰富的形式化语义，是进行跨域整合和智能分析的利器。关联数据和大数据这两个“热词”都是网络数据管理的前沿领域，它们正在带来许多技术突破，为当前信息管理深入到“数据”层面，提供关键性的、令人兴奋的解决方案。

关联数据中的数据并不是独立的、上下文无关的抽象数据，而是一个个具有 URI 标识和 RDF 描述(包括跨域链接) 的明确的知识单元，是 Web 中受管理的基本语义单位。其含义具有特定的上下文关联或明确的元数据描述，能够被计算机所处理。任何一个事物、人物、机构、场所、事件、概念等都可以描述为一条关联数据，因此关联数据向大数据发展是必然的。

越来越多的关联数据应用系统由于数据量越来越大，不得不考虑采用大数据解决方案。由于大数据方案对于海量、高速发展的数据具有很好的管理能力，它被用来管理关联数据是一个必然的选择。这种采用大数据技术方案建立的关联数据应用，可称之为“大”关联数据应用。

越来越多的大数据应用由于采用了 RDF 数据模型进行描述和编码，通过元数据和其他语义描述(本体) ，使数据具有机器可识别的语义(即形式化的语义描述) ，丰富了大数据的语义，并使数据具有更好的互操作性。由于采用语义描述的关联数据已不再是堆砌的信息，结合大数据挖掘手段，还能给大数据分析带来强有力的工具，使大数据集相互之间能够实现基于语义的整合[6]。这种在大数据的应用中采用关联数据技术的应用，可称之为“关联的”大数据应用。

并非所有的关联数据都需要用大数据技术来发布、管理和挖掘，也并非所有的大数据应用都需要利用关联数据等技术进行“语义化”。只有那些数据量大到一定程度，且还在不断膨胀、难以用常规的 RDF 仓储(基于关系数据库或全文检索技术) 提供解决方案的关联数据应用，才需要考虑大数据解决方案; 同样，对于那些开放性和互操作性要求非常高，且在数据挖掘和分析中对于概念的规范性要求非常高，甚至需要提供“知识”挖掘手段的大数据应用，采用关联数据方式提供语义支持才是必要的。

2　关联数据与 NoSQL

2.1　RDF 仓储: 从关系数据库到大数据系统

目前在关联数据的注册站点 CKAN(Comprehensive Knowledge Archive Network) 上登记有 337 个数据集②，根据 2011 年 9 月 19 日的数据分析③，当时的 295 个数据集共包含 316 亿多条三元组，外链数达 50 多亿，其中政府信息是数量最多的应用领域，RDF 三元组有 133 亿多条，FOAF 文件 73 万多个，包含 8.34 亿人名(foaf: person) 。仅仅从维基百科抽取的关联数据 DBpedia 就包含 340 万个概念，10 亿多个三元组，GeoNames 提供了全球 750 万个地理概念(包括地名) 的 RDF 描述，UMBEL 从著名的知识库 OpenCyc 中提取了 2 万个概念类，建立了它们之间的关系链接并指向 DBpedia 和 YAGO 中超过 150 万个实体④，目前各类关联数据应用的数据量已相当惊人。

通常管理 RDF 三元组的仓储系统被称为 RDF 仓储(RDF Store) ，是一种存储、管理和发布 RDF 数据(支持 SPARQL 查询) 的工具，由于 RDF 数据的开放性，一般 RDF 仓储存在分布式环境，特别是面向 Web 的环境中。

RDF 仓储可以建立在传统的关系型数据库(RDBMS) 之上，它用关系数据库作为底层数据管理工具，然后通过 RDB2RDF 方式支持 RDF 数据的生成、发布功能，并具有通常的添加、读取、更新和删除(CRUD) 操作以及 SPARQL 查询等。完全结构化(Schemaful) 的

① W3C 于 1998 年 4 月发布的一项描述网络资源的技术标准，参见 http://www.w3.org/TR/1998/WD-rdf-schema-19980409/，目前被新版标准取代(http://www.w3.org/TR/rdf-syntax-grammar/) 。

② http://datahub.io/group/lodcloud，[2013-02-28]。

③ 参见 http://lod-cloud.net/state/。

④ 数据来自 http://en.wikipedia.org/wiki/Linked_data。

关系型数据库以行的方式存储数据记录，采用 B 树索引，以 SQL 语言查询。依靠服务器性能提高（Scale up，即 CPU 数量性能、内存容量以及总线速度等升级）方式解决海量数据带来的效率问题。

关系型数据库不是管理三元组数据的最好方案，除了 RDF 三元组的数据模型不适宜以关系表的形式存在（即以关系模型表达）之外，关系型数据库对于数据量的支持能力也是一个大问题。通常商用的关系数据库系统（RDBMS）管理的数据量极限在 TB 级，最多不超过 100 个 TB，PB 级的数据量是不可想象的。像 RDF 这种细碎而繁复的数据，其数据之“大”主要体现在数量众多，当数据量达到数亿乃至数十亿条时，RDBMS 在管理效率上基本无法满足一般的管理需求，而且成本高昂。

传统的关系型数据库系统与大数据解决方案的特性对比如表 1 所示：

表 1　RDBMS 与 NoSQL 数据库的对比[7]

比较项	传统的关系数据库系统	大数据/MapReduce 系统
支持数据量	GB 级	PB 级
数据特点	增长量不大	巨量增长
数据更新	读写多次	写一次读多次
数据结构	静态，难以改变	动态，灵活可变
存取方式	交互式	批处理/接近实时
同步要求	高	低
查询语言	SQL	UQL（编程实现，无标准）
完整性要求	很高	非 100%
容错性	单点故障	高容错性
可扩展性	非线性，有极限	可线性扩展
分布式处理	困难	专门为分布式设计
特点	数据管理/报告状态/查找答案	分析结果/发现问题/预测趋势
擅长	事务处理 企业级应用	非结构化数据处理 全网域应用

2.2　NoSQL：大数据解决方案

正在兴起的“大数据”解决方案为解决上述 RDF 仓储的瓶颈提供了新的可能。本文中所称的“大数据解决方案”主要指支持分布式计算的非传统数据库系统，即 NoSQL 数据库。另有一类专门开发的三元组仓储，如 4Store，AllegroGraph，BigData，BigOWLIM，Virtuoso 以及 YARS2 等，采用类似分布式计算的架构（例如 MapReduce）以及图形数据库（NoSQL 的一种）作为底层工具，实现类似的功能。现在这些方法互相借鉴融合，采用 NoSQL 数据库建立 RDF 仓储成为一种趋势，上述三元组仓储也可归入 NoSQL 数据库范畴。

NoSQL 是 Not Only SQL 的缩写，是区别于传统关系型数据库的数据库管理系统的统称。其实 NoSQL 这个词最早于 1998 年出现的时候并非现在的含义，它的发明人 Carlo Strozzi 当时开发了一个轻量级、开源且不提供 SQL（No SQL）功能的关系数据库。由于人们需要的并非“No SQL”，而是“No Relational”，也就是非关系型数据库。

2009 年，NoSQL 的特性被进一步明确为指代那些非关系型、分布式（无法支持联接 Join 运算）、且通常不遵循 ACID 原则的数据库管理系统。所谓 ACID 是传统的关系型数据库系统必须遵循的 4 项原则的缩写，即在数据库管理系统中，事务处理（Transaction）必须具备的 4 个特性：原子性（Atomicity）、一致性（Consistency）、隔离性（Isolation，又称独立性）、持久性（Durability）。

NoSQL 是对传统关系型数据库的扩展，强调的是与“关系型”相对应的“非关系型”特性。NoSQL 通常是半结构化（Schemaless）的数据库，目前数量众多的 NoSQL 数据库大致可分为 4 种类型：键值对（Key - Value Stores）、大表（BigTable）、文档库（Document Stores）和图形数据库（Graph Database），这 4 类数据库支持数据结构的复杂性依次增加，支持“大”数据的能力依次递减，但都能依靠最普通的市售服务器解决扩展性问题（即 Scale out）。

RDF 三元组数据模型从根本上可以看成是图形数据模型的一种特例，最适宜采用图形数据库。图形数据库特别适用于超大量的数据节点以一定的关系链接起来的形式，不管节点内部的数据有多复杂，它都能高效地进行添加、删除、更新和查询操作。而且正由于它对节点内的数据没有限制，在进行大数据分析时往往能得到更多的启发[8]。其他一些大数据系统，例如很多著名的键值对（如 Cassandra）和大表数据库系统，甚至文档型数据库（如 CouchDB）都有对 RDF 数据的内置或扩展的处理功能，都能比关系型数据库更好地解决 RDF 数据的管理和查询问题。

NoSQL 数据库种类繁多，适应性各不相同，但都是传统的关系型数据库的发展和补充，可以看成是对传统关系数据库在某些方面的限制放宽而具有的特殊的扩展能力。当然这些技术相互之间也在借鉴，随着 NoSQL 的发展，甚至有些 NoSQL 数据库（如以 BerkeleyDB 为基础的 Oracle NoSQL 数据库）宣称已经能够

提供 ACID 特性保证。

2.3　NoSQL 作为 RDF 仓储的优势与不足

采用 NoSQL 作为 RDF 仓储通常具有如下优势:

(1) 能够支持灵活的数据结构,支持半结构化甚至无结构的数据存储和管理,特别具有面向对象的数据管理能力。RDF 数据是一种典型的半结构化数据。

(2) 无需数据模式,可以随时任意添加属性(列)而不影响模式。这里的数据模式指定义数据结构的关系型数据库模式,与 XML 模式类似,而不是 RDF 模式,RDF Schema 只是 RDF 术语的扩展,并不定义 RDF 编码的结构。这类数据库系统特别适用于数据模式比较"模糊"的应用,以及采用原型迭代法开发的系统,或者需要被整合到其他模型中的数据。

(3) 具有 NoSQL 带来的高扩展性/伸缩性,同时能以非常低成本的代价(得之于 Scale out 而不是 Scale up 的扩展方式)获得很高的计算性能。这也意味着能够非常容易地支持 RDF 仓储发展为大数据应用,而且特别适合数据量非常庞大的应用,当数据量特别庞大的时候,性能几乎不会受到任何影响。

(4) 支持分布式数据模型,适应网络化的计算环境,特别对于需要基于全网域(Web Scale)提供开放的信息交互、整合和共享时。这一点常常是图书馆等公共信息服务行业的需求。

由于 NoSQL 是新兴技术,目前以开源为主,还缺乏系统完整的技术标准和方案架构,无标准的查询语言,还不够成熟,在市场上缺乏相应的人才和技术支持,也缺乏数据分析和商务智能模型,种类繁多,管理复杂等。这些不足大多能随着技术的完善和应用的普及得到不断改善。

目前关联数据应用需要用到大数据解决方案的还不多,但即便很多应用无需大数据方案的支持,也需要对半结构化数据高效管理、快速检索以及跨域整合的能力,按照数据本来的结构和模式进行管理的 Native 数据库对于数据的完整生命周期管理都具有直接的好处,因此以 NoSQL 取代关系数据库管理关联数据已经成为一个必然的趋势。

3　"大"关联数据的实现

3.1　不同类型的大数据对于关联数据的适应性

大数据有三个基本特性:海量(Volume)、高速(Velocity)和多样(Variety),由于这些特性,大数据应用在性能(Performance)、吞吐量(Throughput)、扩展性(Scalability)和灵活性(Elasticity)4 方面有比较高的要求,但并非所有的大数据应用对这些方面的要求都是相同的。一般而言,大多数的应用只要求一到两个方面,例如社交网络和电子商务的日志文件具有海量和高速的特点,但数据种类并不多;媒体行业数据类型比较复杂,量也很大,然而数据量增长速度并不是非常可观;等等。

对于数量非常庞大的海量数据,目前的分布式数据库技术 NoSQL 和 Hadoop 等都能很好地进行处理,这类大数据应用如果采用语义技术,其挑战主要在于分布式查询(Querying)和推理(Reasoning)方面;对于数据量实时高速增长的大数据应用,这类需求常见于流媒体、传感器等领域,目前分布式计算技术也能很好地满足需求,对语义技术的挑战是流式推理——即在数据源源不断到来的过程中,利用不完全信息进行推理,例如分布式推理可采用 Hadoop/Storm,流式推理采用 Continuous (Dynamic) Queries,把 SPARQL 翻译为 PIG,在 NoSQL 数据库上建立分布式 RDF Store,查询语言有 C - SPARQL, EP - SPARQL,CQELS 等;而对于种类非常繁多、结构十分复杂(异构和半、非结构化数据)但数量可控、变化速度也不是很快的大数据应用,这既是 NoSQL 和 Hadoop 等大数据技术突破传统数据库技术的地方,也是语义技术的强项。如果这三方面要求都很高,目前尚无很好的解决方案。目前大多数应用都要求解决数量的巨大带来的性能问题,一部分要解决数据高速积累时所带来的管理和实时分析问题,三方面都要求很高的应用不多。

3.2　将大数据发布成关联数据

根据关联数据的 4 项原则:数据必须有 URI 标识作为名称、支持 HTTP 访问、提供 RDF 描述以及在 RDF 中尽可能多地提供其他资源的 URI 链接,关联数据系统的开发通常有如下步骤[3]:

(1) 资源命名:能够形式化表达(机器可读)并支持 HTTP 解析。各类 URNs、URIs、URLs 等标识符都可经由命名域转化为 URI,作为资源名称。资源可以多种序列化方式(XML,N3,Turtle 等)表达,也可直接指向另一资源。采用何种序列化方式取决于开发时所采用的平台工具,但由于处理的效率问题,一般推荐 N -

Triple 而不是 RDF/XML 方式序列化。

(2) 领域建模: 即建立领域应用的本体,包括所涉及的、需要描述的实体关系及属性元数据,本体可以是非常简单的、平面的本体,例如 FOAF。本体定义了规范词汇,用作"关系"或取值,可以跨域融合、解析,或本地做语义关系定义(如两个概念的等同,可以定义域内有效)。

(3) 词表映射: 领域模型建立之后要考虑各类实体和关系描述的术语和取值词表。一般应尽可能复用已经成为标准或得到普遍采用的词表,例如可以在 LOD 中已发布的数百个关联数据集中寻找,也可以自己发布成关联数据供别人复用。

(4) 数据转换: 大多数情况下关联数据不是从无到有凭空建设的,总有一些基础数据甚至能够很好地满足企业或机构内部需求的"遗留"系统。这样就可以通过数据转换之间支持将数据导入到关联数据系统中。

(5) 关联建立: 依据领域本体建立起的关联数据系统已经拥有一定的关联关系,但这些关系还需要校验和规整,并且还可以通过 SILK 框架等建立起与万维网上其他关联数据的联系。

(6) 质量控制: 这是关联数据系统能否得到可持续发展的重要步骤,关联数据的意义在于规范性,由于元数据质量和数据转换等过程中产生的问题,必须进行质量控制,并提供一定的错误发现和纠正的反馈机制,使系统越用越好。

(7) Web 发布: 即通过支持各种协商机制的 Web 服务器在网上发布成关联数据应用,通过 API、REST、SOAP 或其他 Web 服务方式,支持 HTTP 查询和关联数据消费。

(8) 搭建 SPARQL Endpoint: 可以作为 Web 发布的一种规范方式,提供统一的服务入口,支持 SPARQL 查询。

(9) 提供 API 接口: 大量的关联数据消费不是来自于浏览器,而是来自于系统后台的应用程序接口,这也是关联数据以机器可理解的语义发布的意义所在。当然本身 HTTP 是通用的关联数据 API 标准,对于企业内部的应用而言,为保护数据资产获取商业利益或提供更复杂的服务,则需要专门的 API 接口。

(10) 使用授权声明: 必须对系统内所拥有的数据和系统所提供的服务的知识产权属性和法律状态做出声明,同时对用户如何使用做出规定。目前对于数据的版权和使用授权已经有一定的规范,可以参照遵循[10]。

将大数据发布成关联数据,基本上也是类似的步骤,核心内容是赋予海量数字对象的单元个体以 URI,将数据格式转化为 RDF,支持与其他关联数据的相互链接。并且系统要对 URI 请求做出正确的响应,能满足事务处理的要求(即通过 REST 或其他方式实现 CRUD)。以下结合大数据特点进行简要说明:

(1) 将 URI 作为资源的唯一标识符发布,支持其作为主键(大数据的主键可以重复),URI 不应动态生成,而要"原生"(Native),并符合 CoolURI 规范,或者支持到内部(局域)标识符体系(如 DOI)的映射;

(2) 支持海量 RDF 数据的处理(主要是 RDF/XML),即支持小数据(Small Chunks)输入,又支持大量数据批处理(装载海量 N – Triples 或 N – Quads 文件);

(3) 直接用 HTTP 存取,即 Web 获取;

(4) 支持逻辑分区,例如通过 Named Graphs 管理数据空间(只有这样才能支持大数据处理);

(5) 提供更新工具,即 HTTP PUT/POST,REST 以及通过 SPARQL 等增删改插入等操作;

(6) 支持各种索引方式。各类不同的 NoSQL 索引方式是不同的,但都有高效的独门秘笈;

(7) 支持简单推理,例如需要进行相等判断(owl:sameAs)等;

(8) 富数据(Rich Format,即各类媒体格式)支持,对于数据内部的结构能够进行识别和处理,这是支持非结构化数据的重要功能;

(9) 高效的图处理(Graph Processing),提供海量数据高速(甚至实时)处理和检索的能力。

3.3　目前的大数据系统对关联数据的支持

大数据提供关联数据解决方案,其核心是支持 RDF 数据的处理。当然全面的支持,包括很多内容,例如数据结构和索引方式对于查询效率影响很大。以 Hadoop 为例,可以存储、索引三元组数据,通过将 SPARQL 编译成 MapReduce 任务支持查询任务,得到查询结果再返回成 Hadoop 文件进行处理和显示[11]。也可以采用 HiveQL 编译 SQL 对 MapReduce 的扁平数据进行批查询甚至批处理。

R2RML 是一个很好的工具，支持对 Hadoop 中的扁平数据文件进行映射，通过虚拟的映射执行 SPARQL 查询，以及将虚拟的数据映射输出为真实的三元组数据。对于 Hadoop 内外的数据、数据簇也能通过类似的方法进行联邦查询，甚至可以支持实时查询。从这里可以看到大数据技术常常需要很多组件协同工作，目前对关联数据进行处理的流程和组件还没有规范，从长远看，如果能实现规范的工作流并通过本体注册系统进行术语及编码的规范则非常理想，这将有助于建立更加灵活一致的架构，使得系统的可维护性、查询效率和数据质量也可以得到明显的提高[12]。有一些能够支持 RDF 数据处理，或者有过开发关联数据应用的大数据系统，以下分别简单介绍：

（1）BigQuery 是 Google 的一项云服务，2010 年末开始支持 RDF 端点发布功能，具有 RDF/N－Triple 存储、处理和检索功能。

（2）Cassandra 是一个大表（BigTable）数据库系统，属于 Apache 开源项目之一，为 Facebook 和 Cisco 等公司所推崇。它有一个存储模块（Adaptor）RDF. rb 支持 RDF 数据的存储，另有一数据处理模块 Brisk 可运行于 Cassandra 之上，提供 RDF 数据的管理和分析功能。

（3）CouchDB 是一个 Apache 开源项目，以 Erlang 语言开发，属于分布式文档数据库，支持 MapReduce 方式查询与索引。它将数据以 JSON 文件形式存放，可用于 Ubuntu 等系统下。它也有一个 RDF. rb 模块，由于原生支持 JSON，关联数据以 JSON－LD 方式是非常合适和高效的。

（4）Hadoop/Pig，Hadoop 是一个使用 Jave 开发的、具有高可靠性和高扩展性的分布式计算框架，也属于 Apache开源项目的一个子项目，Apache Pig 是建立在 Hadoop/MapReduce 框架之上的高层数据分析语言，有多个项目尝试利用 Pig 做 RDF 支持以及 SPARQL 处理，或 RDF 数据批处理模块（例如亚马逊的 Elastic MapReduce）。

（5）HBase 也是 Apache 开源的一个子项目，是与 Google 的 Bigtable 类似的分布式列数据库，使用 Java 开发，应用非常广泛。2009 年就提供了 RDF 数据处理的支持。

（6）MongoDB 也是一个基于 JSON 的文档数据库，使用 C ++ 开发，有一个 MongoDB: : RDF 模块。很多著名网站都使用了 MongoDB，如 Source－Forge（开源软件寄存与下载基地）、CERN（欧洲原子能机构）和 Foursquare 等。

（7）Neo4j 是一个以 Java 开发的、内嵌 RDF 处理功能的图式数据库，包含了索引功能。

（8）SimpleDB 是亚马逊开发的分布式数据库/Web 服务工具，使用 Erlang 开发。常与 AWS 一起配合使用，用户众多，如 Alexa、Netflix 等。Stratustore 项目曾用它成功处理海量 RDF 数据。

（9）Sindice 是一个语义索引工具，在 Hadoop 或 Lucence 环境下具有海量（上亿）三元组的处理能力。

（10）Riak 是一个支持分布式数据管理的键值数据库，内置 MapReduce 支持，具有高可用性，对 RDF 存储和 SPARQL 查询的支持也很好。

（11）微软、甲骨文和 IBM 也都有支持语义数据管理的大数据解决方案，分别是 Connected Services Framework 3.0（微软），Oracle Database Semantic Technologies（甲骨文）和 DB2 NoSQL Graph Store（IBM）。

以上只是不完全的举例。可以看出，目前大数据技术本身还处于“战国时代”，各类套件、解决方案层出不穷，其实几乎所有的大数据工具都能够支持语义数据的处理，差别只在于成熟度、效率的高低以及支持公司的实力。将来处理 RDF 数据极可能成为大数据 NoSQL 数据库的必备选项。

4　“关联的”大数据

4.1　关联数据技术给大数据带来什么？

大数据是超出通常数据库系统处理能力的数据。大数据主要有三个来源：

（1）来自于服务器生成的数据，例如各类日志文件，成百万数量级；

（2）来自于网站的用户，即 UGC，用户创造的信息，例如 Facebook、Twitter 等社会性网络，十亿数量级；

（3）来自于各类数字设备，例如各类传感器、物联网设备、智能手机等，这类数据越来越多，越来越普遍，有上百亿数量级。

这三类数据加上它们的时空信息，构成目前最主要的大数据时空。

大数据也可以根据其产生的来源，分为系统内生成和系统外产生两类，前者主要指事务性数据，分析性

数据，OLTP/OLAP 等结构化数据和文本、各类文件、多媒体(BLOB)和网页等半结构非结构化数据；后者主要指社会性网络数据、传感器数据、政府数据等。

这些无法用传统关系型数据库系统管理的数据，随着大数据技术的逐渐成熟，不断有新的解决方案提出。此时人们很容易把注意力只集中于大数据应用的技术层面，例如查询效率、扩展性、提供实时分析等，而忽视了数据的长期价值和战略作用常常需要有稳定的、统一和整合的、贯穿数据整个生命周期的管理。

关联数据正是这样一种技术。通过关联数据，可以为大数据的术语和属性添加规范名称，进行规范控制，通过发布或复用领域本体，为各类实体建立起本体联系，进而能够使语义联系遍布于整个互联网。它不仅能够作为大数据解决方案的一种补充，提供基于语义的思考框架，而且能在更高层面，从宏观上考虑各数据仓储的相互联系。即既有微观数据之间的链接，又有宏观的本体映射。利用关联数据技术，能够给基于大数据的商务智能带来新的附加价值。只有当大量的数据资源链接成为一个整体视图的时候，才有助于产生新的观察。

4.2　关联数据在哪些方面帮助了大数据？

一般的大数据应用都是基于云计算架构，通常包含以下功能模块：数据收集与发现(Detection)、管理与整合、检索和分析、流程管理、可视化、数据交换和管理支持等。

目前大量的大数据应用在以下方面存在缺陷，可由关联数据所提供的特性加以弥补：

(1) 弥补数据的完整性(Data Integrity)

在应用系统内部，尤其是在不同数据仓储(Silo)之间需要共享数据时，必须设定一些基本的操作原则，特别是命名规范，以及书写、编码规范等。这里有些属于习惯，在各个数据仓储内部具有一定的一致性的基础上，可以通过软件的方式自动地进行归整。

本体是一种常用的方法，用于提供统一的应用模型，更多的时候它作为相互关联的概念形成的词表，使系统内或跨系统的应用能够具有基本的语义互操作性。

(2) 提供整合(互操作)方式 (Integration)

Web Services 的 RESTful 特例成为万维网调用数据的标准方式，每一个操作都指向固定 URI 的数据，类似于 UNIX 操作系统将任何设备都看作是操作系统中的文件一样，把所有的操作都当成 Web 上的 URI，这样有效地统一并简化了数据操作，使操作的关注点集中于数据而不是技术。

(3) 提供数据管理手段(Data Management)

工资单存储于人事数据库中，公司人员的层级关系可能在 ERP 中，而电话号码则在另外的数据库中——可能是外部数据库。这些资源的管理正可以很好地利用到管理数据的特点。

(4) 提供数据复本(Data Replication)

大型数据服务商(或以数据作为核心资产的网络公司，例如 Google，Facebook 等)都非常重视数据的备份、系统的高可靠性和服务的不间断性。很多云架构或大数据软件(如 Hadoop)本身都具有很强的自我复制和备份机制。然而并不是所有的公司都能做到这一点。当以别人的数据作为服务的重要支撑的时候，不得不考虑如果这个服务一旦中断，该如何处置。因此 Resilient(适应性、弹性、可复苏性)是非常重要的。

(5) 提高数据质量(Data Quality)

包含三个方面：

①数据是否正确：名称的准确性是常见问题；

②数据是否完整：例如有时获取的人名数据无法判断唯一性(有姓没名或者有名没姓，或者重名太多)，有时缺乏必备字段(例如出生年月或身份证号之类)；

③数据是否更新：是否为最新数据。

对于关系数据库应用来说检查数据的完整性应该是一个不难的事情，但是对于大数据来说，特别是如果数据来自不同的仓储，就很难保证。

(6) 支持数据迁移(Data Migration)

(7) 提供数据安全及访问控制(Data Security and Access Control)

5　结　语

对数据进行有效管理是图书情报机构的新疆域，也是数据科学带来的新挑战。数据量大、增长迅速、非结构化、知识发现颇为不易，这些问题是大数据和关联数据技术产生的理由，也是这两项技术的目的。近一百年来，图书情报工作随着知识形态的改变，为实现其职业愿景和社会职能，一直在进行着调整和突破，数字技术使知识内容进一步突破了载体的限制，进一步细粒度化和网络化，一方面向“全网域”化发展，另一方

面变成了“大数据”。这两个趋势交汇，更需要图书情报等专门机构提供更加专业的服务，更体现出图书情报机构的价值和优势。

这一趋势给图书情报工作者带来了巨大的挑战，其中最重要的挑战是让他们的工作再一次从管理载体回归到管理知识本身。众所周知，国外从古希腊古罗马到中世纪，国内从春秋战国代直至明清，所有的图书馆或类似机构的管理者都是大学问家。自从进入工业文明时代，社会分工越来越细，知识的生产（印刷）也成了社会化大生产，图书馆学更是成为“书皮学”，图书馆员的工作只限于载体的收集加工流通，基本上与知识内容没有关系。到了今天，知识的内容与载体能够完全分离，图书情报工作者第一次能够借助计算机和网络技术，深入到知识的内容进行处理，给图书情报工作提出了新的要求。国外的专业教育已经开始向培养“数据科学家”和“数据图书馆员”发展。

将来必然有一类图书馆是摆脱了实体形态的存在，为科研、教学以及社会机构的运行决策提供基本的数据服务。这里所说的数据是互联网上给予标识、组织、传输和管理的基本语义（知识）单元，当前可被计算机管理的知识，其最主要的存在形态就是数据，互联网使全世界的知识连为一体而成为“大数据”，而关联数据管理的是知识与知识之间的联系，关联数据技术使人们能够借助计算机“读懂”并处理知识，从而赋予图书情报工作者以强大的“武器”，从事真正的知识服务。

参考文献：

[1] De Wilde P. A Walk in Graph Database [EB/OL]. (2012 - 05 - 05). [2013 - 01 - 08]. http://www.slideshare.net/pierredewilde/a-walk-in-graph-databases-v10.

[2] De Marzi M. Introduction to Graph Databases [EB/OL]. (2012 - 04 - 29). [2013 - 01 - 08]. http://www.slideshare.net/.../introduction-to-graph-databases-1273578.

[3] Hausenblas M, Grossman R, Harth A, et al. Large - Scale Linked Data Processing: Cloud Computing to the Rescue? [EB/OL]. (2012 - 03 - 01). [2013 - 01 - 08]. http://webofdata.files.wordpress.com/2012/03/closer12-processing-lod.pdf.

[4] 刘炜. 关联数据：概念、技术及应用展望[J]. 大学图书馆学报，2011，29(2)：5 - 12. (Liu Wei. Overview on Linked Data: Concept, Technology and Implementation [J]. *Journal of Academic Libraries*, 2011, 29(2): 5 - 12.)

[5] De Wilde P. Small, Medium & Big Data [EB/OL]. (2012 - 09 - 26). [2013 - 01 - 16]. http://www.slideshare.net/pierredewilde/small-medium-and-big-data.

[6] Dimitrov M. Semantic Technologies for Big Data [EB/OL]. (2012 - 09 - 19). [2013 - 01 - 18]. http://www.slideshare.net/marin_dimitrov/semantic-technologies-for-big-data.

[7] Vicknair C, Macias M, Zhao Z, et al. A Comparison of a Graph Database and a Relational Database [EB/OL]. (2013 - 01 - 17). [2013 - 01 - 20]. http://cs.olemiss.edu/ychen/publications/conference/vicknair_acmse10.pdf.

[8] Guzenda L. Realize the Value in Your Big Data with Graph Technology [EB/OL]. (2013 - 01 - 17). [2013 - 01 - 22]. http://www.objectivity.com/event/dbta-webinar-realize-the-value-in-your-big-data-with-graph-technology/.

[9] 夏翠娟，刘炜，赵亮，等. 关联数据的发布技术及其实现——以 Drupal 为例[J]. 中国图书馆学报，2012，38(1)：49 - 57. (Xia Cuijuan, Liu Wei, Zhao Liang, et al. The Current Technologies and Tools for Linked Data: A Case of Drupal [J]. *Journal of Library Science in China*, 2012, 38(1): 49 - 57.)

[10] 张春景，刘炜，夏翠娟，等. 关联数据开放应用协议[J]. 中国图书馆学报，2012，38(1)：43 - 48. (Zhang Chunjing, Liu Wei, Xia Cuijuan, et al. The Open Application Licenses of Linked Data [J]. *Journal of Library Science in China*, 2012, 38(1): 43 - 48.)

[11] Noels S. NoSQL with HBase and Hadoop [EB/OL]. (2010 - 06 - 17). [2013 - 01 - 26]. http://www.slideshare.net/outerthought/nosql-with-hadoop-and-hbase.

[12] Fujitsu. Linked Data: Connecting and Exploiting Big Data [EB/OL]. [2013 - 01 - 28]. http://www.fujitsu.com/uk/Images/Linked-data-connecting-and-exploiting-big-data-%28v1.0%29.pdf.

（作者 E - mail: kevenlw@gmail.com）