一种基于XML的非结构化数据转换方法

杨 晶 周双娥
(湖北大学计算机与信息工程学院 武汉 430062)

摘 要 XML作为半结构化的语言,因其能预先定义标记等优势被普遍应用于非结构化到结构化信息的转换中。利用POI技术把网络上繁杂的非结构化数据转化为XML半结构化数据,把半结构化数据转化为结构化数据,使用户能够简便地查询所需信息。通过实验对SAX,DOM的解析效率进行了对比,实验表明解析相同大小的XML文件,SAX比DOM效率更高,而且此种差距会随着XML文件的增大而逐渐增大。

关键词 大数据,非结构化数据,可扩展标记语言,文档解析技术
中图法分类号 TP391 **文献标识码** A

Method for Unstructured Data Transformation Based on XML Technology

YANG Jing ZHOU Shuang-e
(College of Computer and Information Engineering, Hubei University, Wuhan 430062, China)

Abstract XML, as a semi-structured language, is widely used in converting unstructured information to structured information because of its special characteristic of pre-defined mark. In this work, the complicated unstructured data on the network was converted to XML semi-structured data through POI technology, then the semi-structured data was converted to structured data by parsing XML file through SAX, which would provide convenience for users to search for information. In addition, those efficiencies of parsing of XML files though methods of SAX and DOM were compared in this work for the first time. It demonstrates that the parsing efficiency of SAX is higher than DOM when they are used to parse the same file, and this gap will increase with the size of XML file.

Keywords Big data, Unstructured data, Extensible markup language, Document resolution technology

1 引言

现如今,互联网的普及导致网络上的各类信息数量激增,大数据时代到来。而网络上的大部分信息可以归为office文档、公司报表、图片、音频、视频等类型,这些信息大都是非结构化的[1]。非结构化数据具有样式多样、数据量大等特点,所以用户想通过网络有效找到所需信息变得十分困难。因此过滤掉无用的信息以便用户查询变得越来越重要,这需要利用转换非结构化数据的方法来对网上的信息进行转换。

转换非结构化数据的具体方法有:对超文本标记语言(HTML)进行去标记、分类;制定一些XML模板,例如由文档类型定义(DTD)或者由一种用于描述和规范XML文档逻辑结构的语言(Schema)制定对应的规则,生成XML Schema[2]。施伟斌等[3]通过从XML文件提取结构信息来创建一个临时的DTD,将XML文件映射为对象数据库。李爱民等[4]通过模板建立Schema并将XML文件结构存入其中,再进行解析。Matens等[5]深入研究了DTD,并讨论了XML架构受元素声明一致性(EDC)规则的影响。本文是针对网上图书信息,通过利用SAX(Simple API for XML)解析XML文件的方法把非结构化图书信息转换成结构化数据。SAX是专为解析XML文件而开发的一种方法。转换过程使用POI技术把非结构化图书信息转换为XML文件,再通过SAX解析该文件并导入到数据库。此外,本文使用SAX和DOM算法分别对不同大小的XML文档进行解析,统计其解析效率,以说明SAX方法的高效性。

2 非结构化数据转换为结构化数据的方法

非结构化数据转换为结构化数据的方法大致分为两类:直接转换法和间接转换法,间接转换法即为先把非构化数据转换为半结构化数据,然后再转换为结构化数据的方法。因为直接转换法对数据类型和长度等有限制,它既不能随着数据的扩展而扩展,也不能对扩展以后的信息进行检索,因此间接转化法运用得较为广泛。

间接转换法中的半结构化数据常用XML文件来承载。XML作为数据交换的标准应用在各种Web程序中,可扩展标记语言(XML)是分层格式,用于在万维网的信息交流。XML文档由嵌套元素结构从根元素开始,每个元素都有一个与它关联的标记。除嵌套元素外,一个元素或子元素可以有属性和值[6],这使得非结构化的Web内容能够更好地被用户理解。格式良好的XML文档(它的标签是正确的嵌套)不需要符合特定DTD或者架构[7]。因此,XML是比较适合存储

本文受湖北省统计科研计划重点项目(HB131-32)资助。

杨 晶(1990-),女,硕士生,主要研究方向为非结构化数据、大数据分析,E-mail:395831804@qq.com;周双娥(1965-),女,博士,教授,CCF高级会员,主要研究方向为信息安全、大数据分析。

半结构化的数据，它具有易于扩展的优势，也即扩展过程只需要更改相应的 DTD[4]。

3　解析技术

本文主要采用以 XML 为介质的转换方法，通过 xmlparser 来解析 XML 文件，以实现非结构化数据转换为结构化数据的目的。xmlparser 是一种 XML 的解析器，常见的 xmlparser 有：SAX，DOM，JDOM 和 DOM4J 等，其中 DOM 是 XML 解析技术的基础方法，能使用户很好地理解 XML 文件结构[8]。SAX 因其解析速度快且占用内存少而适用于移动设备，是目前较为流行的解析方法。

3.1　DOM 解析技术

DOM 即文档对象模型，DOM 定义了所有 XML 元素的对象和属性，以及访问它们的方法（接口）。因为 DOM 分析器是以树的形式把 XML 文档放在了内存中[9]，所以 DOM 中每个节点都可通过 Java 或其他编程语言来访问。如图 1 所示，DOM 方法将 XML 文件模拟为一系列节点对象，根元素为 XML 文件根元素，根元素下面的元素节点为网站抓取的书的 div，元素 1—4 为我们要抓取的书的属性。

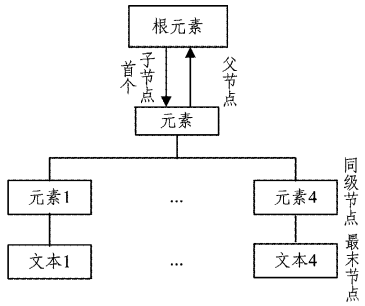

图 1　XML 文件模型

3.2　SAX 解析技术

SAX 属于事件驱动，是应用于实际案例之后产生的标准，不需建文档树。它的原理是使用事件机制和方法回调技术来实现 XML 文档的解析[2]。SAX 应用程序接口（API）中主要有 4 种处理事件的接口[10]，分别是 ContentHandler，DTDHandler，EntityResolver 和 ErrorHandler，表 1 列出了 4 个事件处理器。

表 1　事件处理器

处理器	处理事件
ContentHandler	1. 文档的开始和结束 2. 元素的开始和结束 3. 可忽略的实体 4. 字符 ...
DTDHandler	处理文档 DTD 进行解析
EntityResolver	处理外部实体
ErrorHandler	处理文档解析错误

4 个处理器分别处理不同对象的事件，编译者通过事件处理器提供的方法参数，就可以得到 SAX 解析器解析的数据。XMLreader 接口进行读取分析，DefaultHandler 类是实现以上 4 个事件的处理器接口，XMLreader 读取 XML 文档是调用 ContentHandler 类。ContentHandler 类包含以下几种方法：startElement() 方法、Characters() 方法和 endElement() 方法等[11]，详细调用方法见本文算法部分。

DOM 与 SAX 的解析过程如图 2 所示，XML 文件通过解析器使用两种方法 DOM 和 SAX 对其进行解析。DOM 方法是读入 XML 文档并构建一个树形结构，解析时对内存的要求较高，当文档过大、过于复杂时，对其遍历会相当费时，导致解析效率比较低[12]。SAX 方法是把所有的数据都封装成一个个事件，需要在解析时通知解析器调用相应方法对其解析，因此不需要将整个文档存储之后再进行读取，极大地提高了整个解析过程的效率。两种方法都可以达到解析 XML 文档的目的，但相比内存占用率过大的 DOM，SAX 是较好的选择。

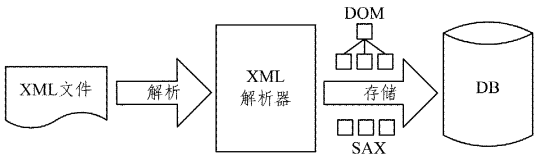

图 2　DOM 与 SAX 的解析过程

4　非结构化数据转化为结构化数据的流程

本文转换过程分为两部分：XLS 文件转换为 XML 文件；XML 文件转换为结构化数据，转换过程的流程如图 3 所示。转换 excel 表的过程是利用函数库（Apache POI）对 excel 表进行读写，POI 提供 API 给 Java 程序对 Microsoft Office 格式档案实现读和写的功能[13]。XML 的解析过程即是对文档进行顺序扫描的过程，当扫描到文档与元素的开始与结束时通知时间处理函数，再由时间处理函数做处理，然后继续同样的扫描，直至文档结束。

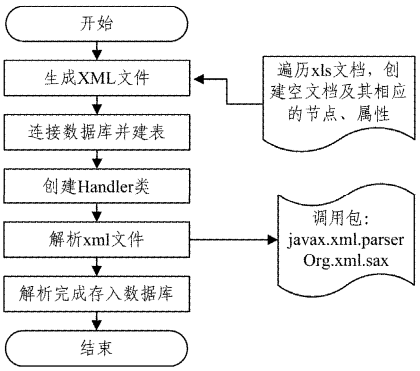

图 3　SAX 解析流程

5　算法描述

5.1　excel 表转化为 XML 文件

5.1.1　算法思想

此算法是通过 POI 函数库提供给 Java 程序的 API 对 excel 文档进行读取。算法中通过遍历 XLS 文件、获取单元格的位置等算法，最终生成 XML 文件并输出。

5.1.2　算法步骤

excel 表转化为 XML 文件的步骤如下。

Step1　创建 XML 文档根元素 Document xmlReport；

Step2　读取 excel 文件并创建 excel 工作表来获得 Excel 工作簿；

Step3　用 rowIndex 遍历工作簿的行；

Step4　迭代每一行时创建 XML 的行元素，获得当前行的单元格数；

Step5　用 cellIndex 遍历行中的每一个单元格；

Step6 获得单元格所在列位置；

Step7 去掉非法字符，根据不同的属性添加属性和元素；

Step8 创建 Serializer，输出 XML 文件。

5.2 SAX 解析 XML 文件

5.2.1 算法思想

此算法是用 SAX 解析 XML 文件并导入数据库。本算法采用直接调用 XMLParser 的方法对文档进行解析，并将每个事件发送给相应的事件处理器。解析过程需要一个类来继承 Android 系统提供的 ContentHandler 类。算法主要分为 4 个部分：接收文档的开始、接收文档的结束、解析一个元素和对字符数据的处理，具体如下：

```
〈sheet〉
  〈row〉//Start Element() 元素开始
    〈name〉标题〈/name〉
    〈author〉作者〈/author〉
    〈pulisher〉出版社〈/pulisher〉
    〈price〉价格〈/price〉
  〈/row〉//End Element() 元素结束事件
  〈row〉
    〈name〉Java 从入门到精通(第 3 版)(附光盘 1 张)〈/name〉
    〈author〉作者：明日科技 编著〈/author〉
    〈pulisher〉出版社：清华大学出版社〈/pulisher〉
    〈price〉￥40.70〈/price〉
  〈/row〉
  〈row〉
    〈name〉Java 核心技术 卷 1 基础知识(原书第 9 版)〈/name〉
    〈author〉作者：(美)〈/author〉
    〈pulisher〉出版社：中国电力出版社〈/pulisher〉
    〈price〉￥80.40〈/price〉
  〈/row〉
〈/sheet〉//End Document() 文档结束事件
```

5.2.2 算法步骤

SAX 解析 XML 文件的步骤如下。

Step1 建立数据库连接。

Step2 开始解析 XML 文件，执行 void startDocument() 接口，调用文档开始事件 SAXException。

Step3 结束解析 XML 文件，执行 void endDocument() 事件回调事件 SAXException。

Step4 开始解析元素，执行 void startElement()事件，输出属名和属性值。结束元素的解析，遇到结束标签时调用 endElement 方法，对标签取值并处理。

Step5 对字符数据进行处理，回调方法 void characters()。

Step6 解析结束导入数据库。

6 实验结果与分析

6.1 实验环境

实验的硬件环境为 intel Core CPU 1.7G Hz，内存为 6G。软件环境为 Microsoft windows7 操作系统。网络爬虫软件为“八爪鱼采集器”。开发平台为 eclips，开发语言为 Java。数据库服务器为 SQL Server 2005。

6.2 数据来源

实验数据数据来源于“当当”图书购物网站关于“Java”关键字的图书信息：书名、作者、出版社、价格。

6.3 实验方法

(1)本文利用爬虫软件抓取书名、作者、价格等图书信息，并用 excel 表对信息进行存储，在 eclips 环境下使用 Java 语言并将 excel 表转换为 XML 文件，其部分 XML 数据如 5.2.1 节中代码所示。

(2)同环境下解析 XML 文件并导入数据库，结果如图 4 所示，其中数据规整、简洁，用户可以快速浏览书目信息。

GHPLJTLRJHZ79DD.dangdang - dbo.当当 GHPLJTLRJHZ79DD...

书名	作者	出版社	价钱
Java从入门到...	作者:明日科技...	出版社:清华大...	¥40.70
疯狂Java讲义...	作者:李刚	出版社:电子工...	¥73.10
Head First Java...	作者:(美) 塞...	出版社:中国电...	¥47.30
Java开发实战1...	作者:李钟尉，...	出版社:清华大...	¥84.80
深入分析Java ...	作者:许令波	出版社:电子工...	¥61.70
Java编程思想...	作者:[美] Bruce...	出版社:机械工...	¥70.20
Java语言程序...	作者:Y.Daniel Li...	出版社:机械工...	¥66.70
Java核心技术 ...	作者:(美) 霍...	出版社:机械工...	¥109.10
Java并发编程...	作者:Brian Goet...	出版社:机械工...	¥54.40
Java Web从入...	作者:明日科技...	出版社:清华大...	¥54.50
Java 学习笔记...	作者:林信良 著	出版社:清华大...	¥53.70
Java网络编程...	作者:(美) 哈...	出版社:中国电...	¥67.90
Java多线程编...	作者:高洪岩	出版社:机械工...	¥54.50

图 4 XML 文件转换结果

6.4 测试方法

分别用 DOM、SAX 解析不同大小的 XML 文件，并比较其解析时间。

6.5 测试数据

测试的 XML 文档来自爬虫软件分别从网站上抓取的 50kB，100kB，150kB，200kB，250kB，300kB 大小的 XML 文件。

6.6 测试结果

本测试由于 PC 机硬盘的反复读取而存在误差。如表 2 所列，解析 56～1076.25kB 大小的 XML 文件，DOM 用时 94～165ms，SAX 用时 77～112ms。

表 2 解析时间

XML 文件大小/kB	DOM/ms	SAX/ms
56	94	77
116	103	87
162	114	88
219	141	92
252	125	93
329	128	95
409	129	99
536	131	102
896	148	106
1076.25	165	112

解析效率 $P=\frac{\text{文件大小}\ m}{\text{解析时间}\ t}$，$P_{DOM}$ 为 0.6～6.52kB/ms。如图 5 所示，P_{SAX} 为 0.73～9.61kB/ms。解析不同大小的 XML 文件 SAX 用时较短，SAX 的解析效率比 DOM 高，并且随着 XML 文件的增大，SAX 与 DOM 解析效率的差距也逐渐增大。这是因为 SAX 解析整个文档无需全部加载到内存中，内存占有量小，解析速度快。

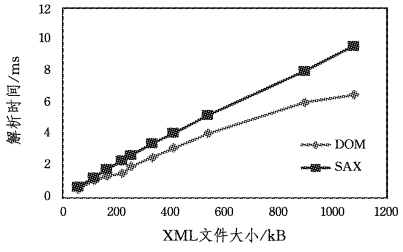

图 5 解析效率对比

结束语　本文将半结构化 XML 文件作为桥梁将网上爬虫的非结构化数据转化成了结构化数据。转换过程利用了 SAX 技术对 XML 文件进行解析，并通过实验对比证实了在 DOM 解析的基础上 SAX 方法较好地提高了解析效率。但本文还有不足之处，只对文档类文件的处理进行了研究，缺少对其他类型文件的研究，在以后的学习中会加强这方面的研究。

参 考 文 献

[1] 万里鹏. 非结构化到结构化数据转换的研究与实现[D]. 成都：西南交通大学，2013.

[2] CHIEW W S, HAW S C, SUBRAMANIAM S, et al. Labeling Schemes for Xml Dynamic Updates: a Survey and Open Discussions[C]// Proceedings of the 2014 International Conference on E-commerce, e-business and E-service. 2014: 79-83.

[3] 施伟斌，孙未未，施伯乐. XML 数据的结构化处理方法[J]. 计算机研究与发展，2002，39(7)：819-826.

[4] 李爱民，谭献海. 基于 XML 技术的非结构化数据到结构化数据转换的研究[J]. 铁路计算机应用，2012，21(10)：12-15.

[5] MARTENS W, NEVEN F, SCHWENTICK T, et al. Expressiveness and complexity of XML Schema[J]. ACM Transactions on Database Systems, 2006, 31(3): 770-813.

[6] SHANMUGASUNDARAM J, SHEKITA E, BARR R, et al. Efficiently publishing relational data as XML documents[J]. The VLDB Journal, 2001, 10(2): 133-154.

[7] 鉴保瑞，宋余庆，陈健美，等. 一种基于关系的 XML 文档模型映射方法[J]. 计算机应用研究，2011(12)：4621-4624.

[8] 冯进，丁博，史殿习，等. XML 解析技术研究[J]. 计算机工程与科学，2009，31(2)：120-124.

[9] 贾福林，王国仁，于戈. 基于 DOM 的 XML 数据库的索引技术研究[J]. 计算机研究与发展，2004，41(1)：175-186.

[10] 赵俊岚. XML 编程中的 DOM 与 SAX 技术[J]. 计算机工程，2004，30(24)：70-72.

[11] 杨治，鞠时光. 基于 SAX 的 XML 数据结构聚簇存储方法[J]. 计算机工程，2008，34(18)：72-74.

[12] COLLADO E M, SOTO M A C, DELAMER I M, et al. Embedded XML DOM parser: an approach for XML data processing on networked embedded systems with real-time requirements[J]. EURASIP Journal on Embedded Systems, 2007, 2008(1): 163864.

[13] 戴维. POI 实现 Excel 的数据导入导出的研究[J]. 科技信息，2013(1)：107.

(上接第 384 页)

[3] WEISS K, TAGHI M K, WANG D D. A survey of transfer learning[J]. Journey of Big Data, 2016, 3(9): 1-40.

[4] LU J, BEHBOOD V, HAO P, et al. Transfer learning using computational intelligence: a survey[J]. Knowledge-Based Systems, 2015, 80(5): 14-23.

[5] BIONDI G O, PRATI R C. Setting parameters for support vector aachines using transfer learning[J]. Journal of Intelligent & Robotic Systems, 2015, 80(12): 295-311.

[6] YING L, LIU B. Application of transfer learning in task recommendation system[J]. Procedia Engineering, 2017, 174(2): 518-523.

[7] OPBROEK V, IKRAM A. Transfer learning improves supervised image segmentation across imaging protocols[J]. IEEE Transactions on Medical Imaging, 2015, 34(5): 1018-1030.

[8] YANG C J, DENG Z H, CHOI K S, et al. Takagi-Sugeno-Kang transfer learning fuzzy logic system for the adaptive recognition of epileptic electroencephalogram signals[J]. IEEE Transactions on Fuzzy Systems, 2016, 24(5): 1079-1094.

[9] CHENG B, LIU M X, ZHANG D Q, et al. Domain transfer learning for MCI conversion prediction[J]. IEEE Transactions on Biomedical Engineering, 2015, 62(7): 1805-1817.

[10] MEI S Y. SVM ensemble based transfer learning for large-scale membrane proteins discrimination[J]. Journal of Theoretical Biology, 2014, 340(1): 105-110.

[11] UGENT J D, BURM M, KINDERMANS P J. Transfer learning of gaits on a quadrupedal robot[J]. Adaptive Behavior, 2015, 23(2): 69-82.

[12] GAO J, FAN W, JIANG J, et al. Knowledge transfer via multiple model local structure mapping[C]// ACM the 14th SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2008: 283-291.

[13] 洪佳明，印鉴，黄云，等. 一种基于领域相似性的迁移学习算法[J]. 计算机研究与发展，2011，48(10)：1823-1830.

[14] 许敏，王士同，顾鑫. TL-SVM：一种迁移学习新算法[J]. 控制与决策，2014，29(1)：141-146.

[15] VAPNIK V. Statistical Learning Theory[M]. John Wiley and Sons, 1998.

[16] ARGYRIOU A, MICCHELLI C A, PONTIL M. When is there a representer theorem? vector versus matrix regularizers [J]. Journal of Machine Learning Research, 2009, 10(12): 2507-2529.

[17] 邓乃杨，田英杰. 数据挖掘的新方法——支持向量机[M]. 北京：科学出版杜，2004.

[18] XIANG E W, CAO B, HU D H, et al. Bridging domains using world wide knowledge for transfer learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(6): 770-783.

[19] BRUZZONE L, MARCONCINI M. Domain adaptation problems: A DASVM classification technique and a circular validation strategy[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010, 32(5): 770-787.

[20] CHANG C C, LIN C J. LIBSVM: a library for support vector machines[J/OL]. http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.