

## 基于自动机理论的 PDF 文本内容抽取

王晓娟<sup>1,2\*</sup>, 谭建龙<sup>2</sup>, 刘燕兵<sup>2,3</sup>, 刘金刚<sup>1,2</sup>

(1. 首都师范大学 计算机科学联合研究院, 北京 100037; 2. 中国科学院 计算技术研究所, 北京 100190;  
3. 中国科学院 研究生院, 北京 100049)

(\* 通信作者电子邮箱 wangxiaojuan@software.ict.ac.cn)

**摘要:** 现有的从 PDF 文档抽取文本内容的方法(如 PDFBox 类库采用的方法)处理速度较低,无法满足高速网络中内容分析的需求,也不能对网络中部分到达的 PDF 数据包进行流式的处理。为此,提出了基于自动机理论的 PDF 文本内容抽取方法。该方法通过建立具有层次的关键字自动机,可以快速抽取完整 PDF 文档和不完整 PDF 文档中的文本内容。在中文和英文 PDF 文档数据集下的实验结果表明,基于自动机理论的 PDF 文本内容抽取方法耗时仅为 PDFBox 方法的 17%~37%。

**关键词:** 文本内容抽取; 自动机; 确定的有穷自动机; 不完整文档

**中图分类号:** TP311.52 **文献标志码:** A

### Extraction of text content from PDF documents based on automaton theory

WANG Xiao-juan<sup>1,2\*</sup>, TAN Jian-long<sup>2</sup>, LIU Yan-bing<sup>2,3</sup>, LIU Jin-gang<sup>1,2</sup>

(1. Joint Faculty of Computer Scientific Research, Capital Normal University, Beijing 100037, China;

2. Institute of Computer Technology, Chinese Academy of Sciences, Beijing 100190, China;

3. Graduate University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** The existing methods of extracting text content from a PDF file, such as the one adopted by the PDFBox library, are not efficient enough to handle the high-speed network traffic. Moreover, these methods cannot extract the contents streamingly from partial PDF packets in transfer. This paper proposed a new method based on automaton theory. The method adopted a hierarchical keyword Deterministic Finite Automaton (DFA) to extract information from complete or incomplete PDF files. The experimental results show that the response time of the proposed method is about 17% - 37% of the algorithm used by PDFBox when processing PDF files in Chinese or English.

**Key words:** text content extraction; automaton; Deterministic Finite Automaton (DFA); incomplete document

## 0 引言

PDF 文档<sup>[1]</sup>内容抽取是指对 PDF 格式文档进行结构分析,并抽取其中重要的文本内容信息和图片信息的过程。PDF 文档内容抽取的应用范围非常广泛,如学术搜索、商品搜索、文本挖掘、知识库建立以及网络内容过滤。现有的 PDF 文本内容抽取方法处理速度较低,无法满足高速网络中内容过滤和分析的需求,同时也不能对网络中部分到达的 PDF 数据包进行流式的处理,更不能抽取不完整的 PDF 文档中的文本内容。本文实现了基于自动机理论的 PDF 文档内容抽取。

关于 PDF 文档及其应用的研究可分为两类:1) PDF 阅读器的设计。即通过对 PDF 文档的格式进行分析,用面向对象的方法,完成中文 PDF 阅读器的设计与实现<sup>[2-3]</sup>。2) PDF 文档内容抽取。PDF 文档内容抽取又可以分为两大类。一类是通过分析 PDF 文档的格式,直接将其中的文本信息和图像信息抽取出来<sup>[4-10]</sup>。William 等<sup>[7]</sup>研究了 PDF 文档的文件理解和文件分类。首先将 PDF 文档的每页分解成一些几何元素,然后通过分析几何元素之间的关系,将几何元素组成一些逻辑块。分块完成后用改进的 Blackboard 算法分析整个文档的架构并进行归类(如报纸、杂志、宣传册、商业信函等)。Yuan

等<sup>[8]</sup>通过对 PDF 文档的分析,在原来的文本信息中加入一些额外的统一标签后将原来的文本转化成一种半结构化的信息,在这种信息上利用模式串匹配算法得到 PDF 文档题目、作者、地址、摘要和关键字等信息。Chao 等<sup>[9]</sup>对怎样将 PDF 文档的布局以及其中的各种内容抽取出来进行了相关研究。另一类 PDF 文档内容抽取方法是将原 PDF 文档转换为其他文档格式,从而利用抽取中间文档格式内容的方法抽取 PDF 文档中的内容。已有的方法包括基于 XML 的 PDF 文档信息抽取,通过设计科技论文的 DTD 文档,把以 PDF 格式表示的科技论文解析转换为有效的 XML 文档<sup>[11]</sup>;或者基于 XSLT 的 PDF 论文元数据的抽取,它以 XSLT 作为信息抽取规则,利用文本特征、位置特征以及显示特征对中间 XML 文档进行基于 XSLT 规则的信息抽取<sup>[12-13]</sup>。

现有的关于 PDF 内容抽取的方法的缺点主要有:1) 可移植性不好,无法自适应不同版本的 PDF 格式,当文件格式发生变化时,必须修改已有的程序;2) 可扩展性不高,无法对高速网络中的 PDF 数据进行实时抽取;3) 过分依赖文档的完整性,不能对网络中部分到达的 PDF 数据进行流式处理,也不能抽取不完整 PDF 文档中的文本内容。

基于自动机理论的 PDF 文档内容抽取,首先,将 PDF 文

收稿日期:2012-02-21;修回日期:2012-06-23。

基金项目:国家自然科学基金资助项目(61070026);国家 863 计划项目(2011AA010705)。

作者简介:王晓娟(1985-),女,内蒙古赤峰人,硕士研究生,主要研究方向:信息内容安全、模式串匹配;谭建龙(1974-),男,湖南长沙人,研究员,博士,主要研究方向:自然语言处理、网络安全、模式匹配;刘燕兵(1981-),男,湖北麻城人,助理研究员,博士研究生,主要研究方向:信息内容安全、模式串匹配算法;刘金刚(1963-),男,辽宁铁岭人,教授,博士,主要研究方向:操作系统、智能接口。

档中标识文档格式的关键字提取出来,组成关键字树。然后,基于自动机理论将关键字树建立成具有层次的自动机。最后,扫描网络中的数据流或者待抽取的本地文件,识别出其中的 PDF 文档,并将其中的文本信息抽取出来。本文方法不但可以抽取完整 PDF 文档,还可以处理部分破损的 PDF 文档中的文本信息。同时,在处理网络中的数据流时,不需要存储所有的数据,节约了大量内存空间。最后,通过过滤掉一些与抽取文本信息无关的内容,大幅度提高了处理速度,减少了响应时间。

## 1 PDF 文档的结构

结构化的文档格式 PDF<sup>[1]</sup>是由美国排版与图像处理软件公司 Adobe 于 1993 年首次提出的。它由页面描述语言 PS (PostScript) 发展而来,具有与 PS 几乎相同的页面描述能力和相似的描述方法。但与 PS 不同的是,PDF 除了能描述复杂版面外,还具有交互功能(如超链接、交互表单等)、页面随机存取及字体仿真描述等特性。因此,PDF 不仅适合印刷出版,而且也适合电子出版,所以现在很多文档都采用 PDF 格式。

PDF 的结构可以从文件结构和逻辑结构两个方面来理解。其中,文件结构是指其文件的物理组织方式,逻辑结构则是指其内容的逻辑组织方式。

PDF 的文件结构(即物理结构)包括四个部分:文件头、文件体、交叉引用表和文件尾。文件头(Header)指明了该文件所遵从 PDF 规范的版本号,它出现在 PDF 文档的第一行。文件体(Body)由一系列的 PDF 间接对象组成,这些间接对象构成了 PDF 文档的具体内容,如字体、页面、图像等。交叉引用表(Cross Reference Table)则是为了能对间接对象进行随机存取而设立的一个间接对象地址索引表。如果是线性 PDF 文件,则可能有多个交叉索引表,而每个交叉索引表的末尾将给出下一个交叉索引表的相对偏移量。文件尾(Trailer)声明了交叉引用表的地址,指明文件体的根对象(Catalog),还保存了加密等安全信息。根据文件尾提供的信息,PDF 的应用程序可以找到交叉引用表和整个 PDF 文档的根对象,从而控制整个 PDF 文档。

PDF 文档的结构反映了文件体中间接对象间的等级层次关系。PDF 文档是一种树型结构。树的根节点就是 PDF 文档的根对象(Catalog)。根节点下有四个子树:页面树(Pages Tree)、书签树(Outline Tree)、线索树(Article Threads)和名字树(Named Destination)。

## 2 PDF 文档内容抽取方法的设计和实现

### 2.1 功能设计

PDF 文档内容抽取是指对 PDF 格式文档进行结构分析,并抽取其中重要的文本内容信息和图片信息,为以后的内容分析和扫描提供基础的数据。如图 1 所示,PDF 文档的结构分析主要包括:对二进制文档按照文档格式规范进行数据区域分割,形成分段数据;再将每段数据进行解压缩,形成普通数据(如果数据是加密的,还需要在解压前,对分段数据进行解密);然后,对普通数据进行编码转换,形成正常编码数据。

PDF 文档内容抽取技术的核心难点在于文档的格式分析,因为格式分析需要依据不同的文档规范,进行不同的实现。各个厂家为了支持新的功能,同一个格式文档的文档规范也不断地修改,增加新的功能。出于保密的原因,部分文档

格式的定义是部分公开的甚至是完全不公开的。因此实现通用的 PDF 文档内容抽取是比较困难的。

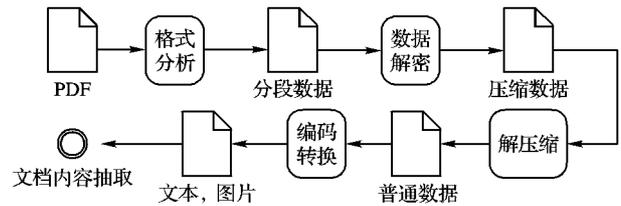


图 1 PDF 文本内容抽取数据流程

文档格式抽取中还涉及到数据内容的解密和解压缩等开销较大的操作。为了提高内容抽取的性能,理论上只需对必须抽取的分段内容进行解码和解压缩,但是如果采用格式文档厂家的编程接口,一般情况下会对整个文档进行解密和解压缩,没有针对内容抽取的特定要求进行特定的解密和解压缩。本文实现了基于自动机理论的 PDF 文档内容抽取技术。

### 2.2 建立关键字树

PDF 文档由以下内容组成:页面、字体、文字布局、图像、表单、表格、多媒体页面等。这些内容都可以抽象地表示为一个二元组,二元组的形式为{关键字,操作},其中操作是关键字定义的处理与其相联的实体内容的动作。在抽取文本内容时,需要找到与内容相关的关键字,根据关键字所定义的动作,来执行相关的抽取工作。

首先将关键字按是否具有层次分成两类。其中,具有层次的关键字可以组织成如图 2 所示的树状结构,在抽取过程中要用分层确定的有穷自动机(Deterministic Finite Automaton, DFA)的实现方法。在关键字组织图中,每一个节点的子节点都是父节点要处理的内容,而要抽取的文本内容位于叶子节点上。

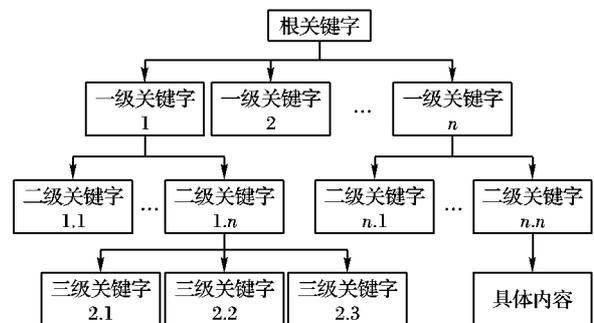


图 2 关键字组织

在 PDF 文本内容抽取的过程中,主要处理几种信息:页面信息、编码解码信息、内容信息和字体信息。页面信息中的关键字有“Pages”、“Page”。“Pages”定义了本文档中的页面结构,通过该关键字,可以建立起页面树。“Page”关键字定义了文档中每页中的详细信息。编码解码信息的关键字是“Filter”,当找到该关键字时,就知道相应的内容流应该用哪种方法进行解码。一个内容流可能使用了多种编码,在解码的过程中,要按照编码的逆序来依次解码相应的内容流。内容信息的关键字是“Contents”,在一个页面信息中,如果有此关键字,说明本页中有文本内容,那么就需要做内容抽取的操作,否则可以直接忽略本页。字体信息的关键字是“Font”。在文本抽取中,需要通过字体信息,来确定内容的文本对象应该在做哪些操作之后才能将内容存储到用户可理解的文本文件中。

### 2.3 分层构造 DFA

本文用基于 DFA 的方法来抽取 PDF 文档中的文本信息。

PDF 文档是有层次结构的,利用 2.2 节建立的关键字树,通过分层建立 DFA(如图 3 所示)的方式,最终将文件中的文本信息抽取出来。

图 3 所示的是抽取 PDF 文档信息的顶层 DFA。顶层 DFA 要完成的工作是识别 PDF 文件逻辑结构中每一部分的开始和结束,当检测到一个部分的开始时,就调用下层 DFA 来具体实现抽取不同内容的操作。操作完成后,返回一个命令通知上层 DFA 操作已完成,上层 DFA 的操作继续进行。

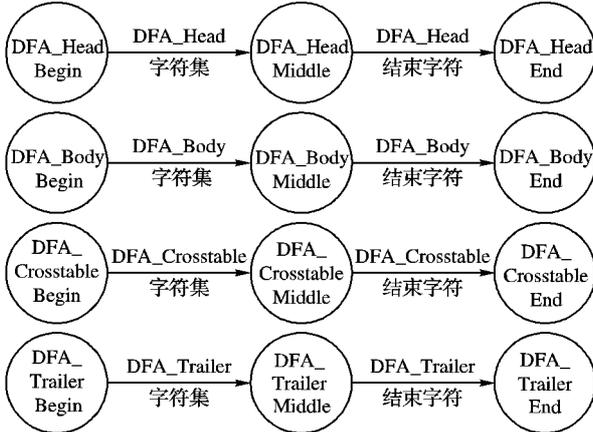


图 3 识别 PDF 文件结构每部分开始和结束的自动机

逻辑结构中的每部分要完成的操作不同,所有在下层 DFA 中,每一部分的操作要具体定义。图 4~7 是识别文件不同部分的 DFA。文件头部分主要是识别 PDF 文档的版本号,版本不同,在解析文本信息时所要做的有些操作也不同。文件体中包括了所要解析的所有内容,这些内容都是由间接对象组成。每个间接对象的格式相同,但具体内容不同。在遇到间接对象时,通过查找 {关键字,内容} 二元组,来确定具体要执行的操作。交叉索引表存放了本 PDF 文件中所有间接对象的位置信息,通过交叉索引表,可以快速找到每一个间接对象。在文件尾 DFA 中,需要找到交叉索引表的开始位置以及加密信息。

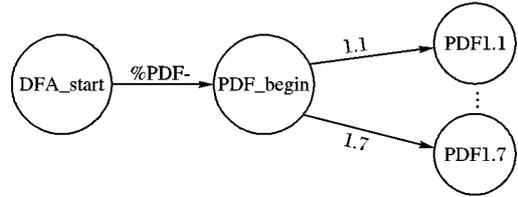


图 4 文件头部自动机

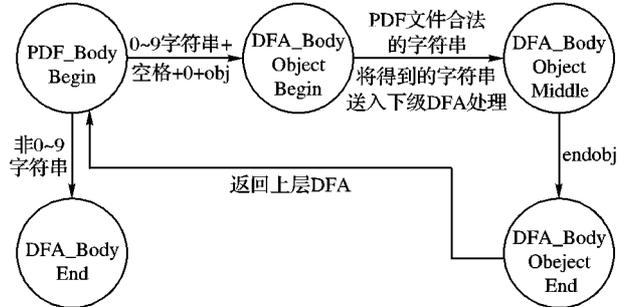


图 5 文件体自动机

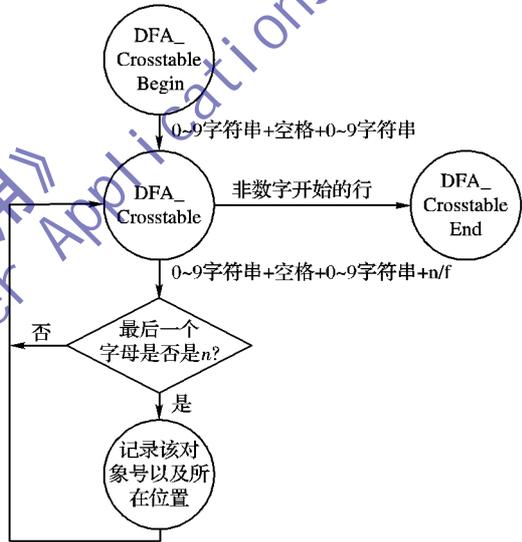


图 6 文件交叉引用表自动机

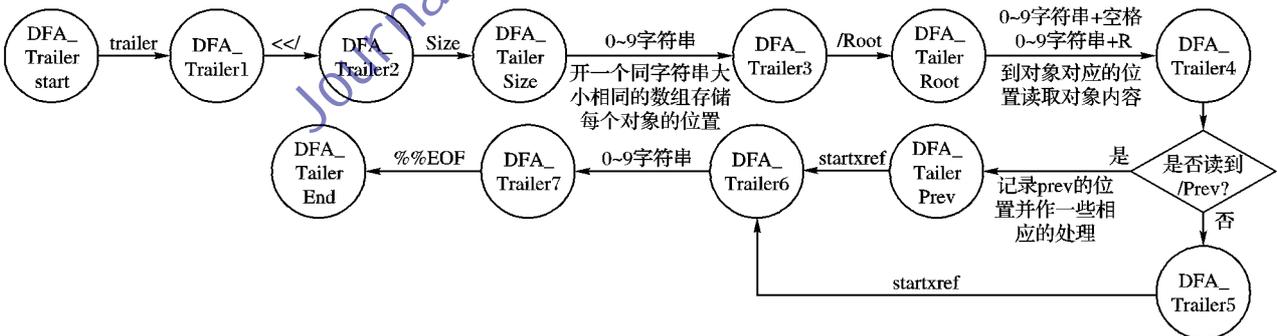


图 7 文件尾自动机

本文用 AC(Aho-Corasick) 自动机<sup>[14]</sup> 建立 DFA,下面是建立识别 PDF 文件头 DFA 的过程。识别文件头的关键字是“%PDF-”,相应的操作是根据 PDF 版本的不同,驱动不同的抽取过程。图 4 是状态转换图,DFA\_start 是整个 DFA 的开始状态,当遇到“%PDF-”时转到 DFA\_Head 开始状态,继续识别输入的字符串。如果接下来的字符串是“1.1”~“1.7”,就转到不同的 DFA\_Head 结束状态。DFA\_Head 的结束状态也是识别 PDF 文件体的开始状态。图 4~7 所示的 DFA 建立过程的方法同图 4,本文不再扩展介绍每一部分 DFA 的建立过程。

基于自动机理论的抽取方法,当文档格式发生变化时,只需在关键字树中添加或删除一些节点,重新生成 DFA,而不需修改源程序,因此本文算法有较好的适应性。该方法把源文档看成一个二进制字符流,所以可以流式地处理网络中部分到达的数据包。对于不完整的文档,也可以通过预先定义好的关键字树进行分析。

### 2.4 文本内容的抽取

本文抽取 PDF 文档内容的过程主要有以下几个步骤:1) 配置一个 {关键字,操作} 的二元组文件;2) 将二元组中的关键字构建成关键字树;3) 将关键字树分层抽取出来,将每

层的所有关键字按 PDF 文件的逻辑结构划分开来,用 AC 算法建立不同的 DFA;4) 自动机建立好以后,就可以扫描网络中的数据流或者完整的 PDF 文档,来抽取 PDF 文件中的文本信息。图 8 是抽取页面中的文本信息的具体流程,其中方法 1 是非 type0 的字型取得 Unicode 码的方法,方法 2 是 type0 字型取得 Unicode 码的方法。具体做法详见文献[1]。

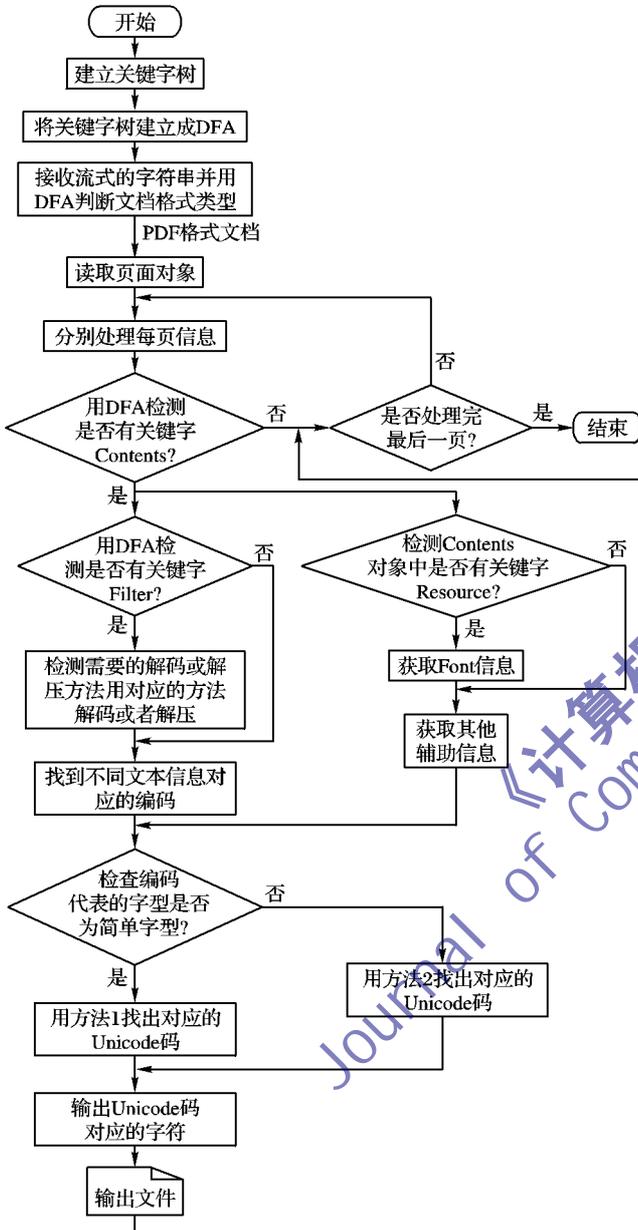


图 8 抽取页面中文本信息的具体流程

### 3 实验结果与分析

将基于自动机理论的 PDF 文本内容抽取方法和 PDFBox 方法进行了对比测试。Apache PDFBox (<http://pdfbox.apache.org/>) 是 Java 实现的开源的处理 PDF 文档的类库,它可用来创建 PDF 文档、处理已有的文档以及从文档中抽取内容。PDFBox 还包括很多命令行实用工具,PDFBox 是抽取 PDF 文档内容时最常使用的一个类库。

#### 3.1 实验数据和环境

本文选取了两类测试数据:中文 PDF 文档和英文 PDF 文档。本文随机地从不同的中文电子书中选取了文件大小不同的中文 PDF 文档,文件大小从 139 KB 到 48 500 KB;英文 PDF

文档是从计算机方面的全文数据库网站或者英文电子书中随机选取的,文件大小从 219 KB 到 1 250 KB。

实验的软硬件环境如下:

CPU: Intel Duo E7500 2.93 GHz 双核;内存: 2.0 GB; Cache: 一级数据缓存 64 KB,一级指令缓存 64 KB,二级缓存 3 MB;操作系统: Windows 7 Professional 32 位。

#### 3.2 实验结果和分析

基于自动机理论的 PDF 文本内容抽取方法和开源库 PDFBox 抽取英文 PDF 文档内容的实验结果如表 1 所示,抽取中文文档的实验结果如表 2 所示。实验中的中文文件来自于一些 PDF 版的电子书,英文文件既有 PDF 版电子书的某些章节,也有国外数据库中的论文。

表 1 英文 PDF 文档抽取结果对比

文件编号	文件大小/KB	抽取时间/ms	
		PDFBox 方法	自动机方法
1	139	437	141
2	462	531	187
3	898	546	203
4	1 300	1 500	468
5	12 500	9 641	2 074
6	48 500	58 496	8 721

表 2 中文 PDF 文档抽取结果对比

文件编号	文件大小/KB	抽取时间/ms	
		PDFBox 方法	自动机方法
1	219	452	78
2	323	515	125
3	468	827	328
4	1 020	1 580	281
5	1 210	807	222
6	1 250	874	312

在中文和英文 PDF 文档数据集下的实验结果表明:本文方法所用的时间仅为 PDFBox 方法的 17% ~ 37%,在实验对比表格中,本文方法所用的时间是构建关键字树、建立自动机和抽取文本内容三部分所用的时间和。本文方法构建的关键字树和自动机可以重复使用,所以在抽取网络中的 PDF 文本内容时,不需要重复构建关键字树和自动机,实际的处理时间就是扫描文件并将其文本内容抽取出来所用的时间。因此在处理网络中的 PDF 文档时,本文方法的处理速度会更快。

在中文 PDF 文本内容抽取中,首先要将每页的具体内容和字体信息进行解码,然后通过 2.2 节所述的相应方法将十六进制表示的字符信息转换成可显示的字符,并存储到文本文件中。

表 3 本文方法抽取英文 PDF 文档内容时各部分所用时间 ms

文件编号	预处理时间	解码时间	输出文本信息的时间	总用时
4	109	218	125	468
5	967	998	190	2 074
6	3 775	3 822	124	8 721
7	998	1 264	374	2 652

从表 3 可看出,在英文 PDF 文本信息抽取中,因为英文 PDF 文档的字符集大多数采用 Adobe 公司提供的 14 种标准字符集,这些字符不需要查询字体信息就可以直接输出,所以

处理时间主要花在解码上。本文方法可以通过分析PDF文档中哪些内容与抽取文本信息无关来尽量减少不必要的解码时间,从而加快处理速度。

#### 4 结语

随着PDF格式被国际标准化组织ISO接纳为国际标准,更多电子文档格式将采用PDF格式,网络中会有越来越多的文件采用PDF格式进行传输。为了对网络中以PDF格式传输的内容进行分析过滤,必须先将二进制的PDF文档中的内容抽取出来。基于以上需求,本文提出并实现了基于自动机理论的PDF文本内容抽取方法。该方法可以快速、准确地将PDF文档中的文本信息抽取出来。本文用预定义的关键字构建DFA,通过精确串匹配算法将相应的关键字查找出来,然后用与关键字相对应的操作处理其中的内容。该方法有较好的扩展性和自适应性,当PDF文档的格式发生变化时,只需修改{关键字,内容}二元组,并重新构建DFA就可以将新的内容加入原有的程序中。当需要对其中的内容进行在线分析和处理时,就可用本文的方法先将内容抽取出来。本文只实现了对几种简单字体的文本信息抽取,未来的研究方向是支持所有字体和版本的PDF文本内容抽取,并且将匹配关键字的方法扩展到正则表达式。

#### 参考文献:

- [1] Adobe Systems Incorporated. PDF reference: sixth edition [EB/OL]. [2010-10-23]. [http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf\\_reference\\_1-7.pdf](http://www.adobe.com/content/dam/Adobe/en/devnet/acrobat/pdfs/pdf_reference_1-7.pdf).
- [2] 杨道良. 面向对象的中文PDF阅读器的设计与实现[J]. 计算机应用, 1999, 19(6): 1-4.
- [3] 李强, 刘时进. PDF阅读器的设计与实现[J]. 计算机工程与设计, 2010, 31(7): 1635-1638.

(上接第2471页)

$$c \equiv d_1 b_1 m + d_0 b_1 + x \equiv a_1 b_1 m^2 + (a_1 b_0 + b_1 a_0) m + a_0 b_0 \pmod{N} \quad (10)$$

且有  $0 \leq c < 2N$ , 经第16)步同余调整后得  $c = ab \pmod{N}$ 。这两步也无溢出。

综合上面的分析,可得定理结论成立。证毕。

#### 4 结语

利用本文算法,在VC++ 6.0版本的编译环境下使用64位整数类型\_\_int64,可以将模的范围从31位的整数提高到62位(考虑补码中符号要占一位)<sup>[6]</sup>并通过了大量测试。该算法也在整数分解的Pollard p-1算法、求离散对数的小步大步算法等数论算法<sup>[2-5]</sup>中得到了实际使用。如果需要更大的模的模乘运算,则需要使用高精度的四则运算。

致谢 特别感谢沈祖和教授在撰写本文前后的鼓励和指导。

#### 参考文献:

- [1] HARDY G H, WRIGHT E M. 数论导引[M]. 张晓明, 张凡, 译. 北京: 人民邮电出版社, 2008.
- [2] CRANDALL R, POMERANCE C. Prime numbers: A computational perspective[M]. Berlin: Springer-Verlag, 2001.

- [4] 李贵林, 李建中, 杨艳. Plug-in实现对PDF文件的信息提取[J]. 计算机应用, 2003, 23(2): 110-112.
- [5] 李珍, 田学东. PDF文件信息的抽取与分析[J]. 计算机应用, 2003, 23(12): 145-147.
- [6] 张秀秀, 张立峰. PDF文件文本内容提取研究[J]. 科技情报开发与经济, 2008, 18(36): 118-120.
- [7] WILLIAM S L, DAVID F B. Document analysis of PDF files: methods, results and implications[J]. Electronic Publishing Origination Dissemination and Design, 1995, 8(2/3): 207-220.
- [8] YUAN FANG, LIU BO, YU GE. A study on information extraction from PDF files[C]// ICMLC 2005: Proceedings of the 4th International Conference Advances in Machine Learning and Cybernetics, LNCS 3930. Berlin: Springer-Verlag, 2005: 258-267.
- [9] CHAO HUI, FAN JIAN. Layout and content extraction for PDF documents[C]// DAS 2004: Proceedings of Document Analysis Systems, LNCS 3108. Berlin: Springer-Verlag, 2004: 213-224.
- [10] TAMIR H, ROBERT B. Intelligent text extraction from PDF documents[C]// CIMCA/IAWTIC 2005: Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Washington, DC: IEEE Computer Society, 2005: 2-6.
- [11] 宋艳娟, 张文德. 基于XML的PDF文档信息抽取系统的研究[J]. 现代图书情报技术, 2005, 21(9): 10-13.
- [12] 陈俊林, 张文德. 基于XSLT的PDF论文元数据的优化抽取[J]. 现代图书情报技术, 2007, 23(2): 18-23.
- [13] 宋艳娟, 李金铭, 陈振标. 基于XSLT的PDF信息抽取技术的研究[J]. 计算机与数字工程, 2008, 36(5): 156-159.
- [14] GONZALO N, MATHIEU R. Flexible pattern matching in strings: practical on-line search algorithms for texts and biological sequences[M]. Cambridge: Cambridge University Press, 2002: 49-54.

- [3] 颜松远. 计算数论[M]. 2版. 杨思嫒, 刘巍, 齐璐璐, 等译. 北京: 清华大学出版社, 2008.
- [4] 裴定一, 祝跃飞. 算法数论[M]. 北京: 科学出版社, 2002.
- [5] 李继国, 余纯武, 张福泰, 等. 信息安全数学基础[M]. 武汉: 武汉大学出版社, 2006.
- [6] 朱怡健, 朱敏, 王健. 计算机组成原理[M]. 南京: 东南大学出版社, 1994.
- [7] MONTGOMERY P L. Modular multiplication without trial division[J]. Mathematics of Computation, 1985, 44(170): 519-521.
- [8] 蒋晓娜, 段成华. 改进的蒙哥马利算法及其模乘法器实现[J]. 计算机工程, 2008, 34(12): 209-211.
- [9] BLAKLEY G R. A computer algorithm for calculating the product AB modulo M[J]. IEEE Transactions on Computers, 1983, C-32(5): 497-500.
- [10] CHEN CHIENYUAN, CHANG CHINCHEN. A fast modular multiplication algorithm for calculating the product AB modulo N[J]. Information Processing Letters, 1999, 72(3/4): 77-81.
- [11] KARL H. Fast division of large integers: a comparison of algorithms[EB/OL]. [2011-03-23]. <http://www.treskal.com/kalle/ex-jobb/original-report.pdf>.
- [12] 徐国良. 任意高精度四则运算的算法与实现[J]. 数值计算与计算机应用, 1983, 4(4): 229-240.