计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(12):18−24 [doi: 10.15888/j.cnki.csa.006675]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

油气行业垂直搜索引擎关键问题解决方案①

王　督, 蔡永香, 李博涵, 刘远刚

(长江大学 地球科学学院, 武汉 430100)
通讯作者: 蔡永香, E-mail: caiyx2002cn@126.com

摘　要: 垂直搜索引擎构建是搜索领域的热点问题之一, 应用领域广泛. 现有的方法一般都只是对垂直搜索引擎构建中的某一个或几个阶段进行优化, 且针对不同网站信息的获取往往需要人工配置操作, 较为繁琐. 本文在深入研究构建垂直搜索引擎技术的基础上, 运用 Heritrix、Solr 等 JAVA 开源工具, 结合网页正文抽取和完整性词抽取算法, 提出了一套自动化构建垂直搜索引擎的方法, 对该方法实现各阶段的关键问题展开了研究, 并给出相应的优化方案. 实践表明, 提出的方法与优化方案具有较强的实用性.

关键词: 垂直搜索引擎; 信息爬取; 网页正文抽取; 完整词抽取; Heritrix 和 Solr

引用格式: 王督,蔡永香,李博涵,刘远刚.油气行业垂直搜索引擎关键问题解决方案.计算机系统应用,2018,27(12):18–24. http://www.c-s-a.org.cn/1003-3254/6675.html

Critical Problems and Solutions for Vertical Search Engine in Oil and Gas Industry

WANG Du, CAI Yong-Xiang, LI Bo-Han, LIU Yuan-Gang

(School of Geosciences, Yangtze University, Wuhan 430100, China)

Abstract: Vertical search engine has always been a hotspot in the study of searching technique. Dispite a wide range of applications, the mainstream method of vertical search engine still has several flaws. In many cases, only a few stages have been optimized in the construction process of vertical search engine. Also, when obtaining information from websites, most of the methods require manual configuration, which is cumbersome. Based on an in-depth study of the vertical search engine technology, this article presents a method that uses JAVA open source tools such as Heritrix, Solr, combined with the extraction algorithm of web content and integrity word for automatically constructing a vertical search engine. In addition, the article examines the key issues in the various stages of the method's implementation and puts forward the corresponding optimization plan, which are examined to have strong practicality.

Key words: vertical search engine; information crawling; webpage text extraction; full word extraction; Heritrix and Solr

随着互联网普及率的不断提高, 网络信息呈现出数量大、种类多、来源复杂, 具有不一致性和不完整性等特点[1]. 垂直搜索引擎是为了帮助用户从海量信息库中快速而准确地获取所需的内容.

目前建立垂直搜索引擎方面的研究主要包括网页数据爬取、网页结构化数据抽取、网页主题性判断和分词和索引策略等方面. 国外对垂直搜索引擎的研究起步较早, 经典的 Fish-search 算法[2]根据 potential-score 值动态地改变抓取队列中 URL 项的顺序, 实现相关网页的快速抓取. 文献[3]在网页采集的过程中, 通过 URL 的 MD5 摘要计算, 避免对相同的 URL 执行多次网页抓取过程. Ricardo[4]等利用分布式的查询代理

① 基金项目:地理信息工程国家重点实验室基金项目 (SKLGIE2017-M-4-6);国家自然科学基金青年基金项目 (41701537);大学生创新项目 (201810489071)
Foundation item: State Key Laboratory of Geo-information Engineering (SKLGIE2017-M-4-6); Young Scientists Fund of National Natural Science Foundation of China (41701537); Graduates Innovation Program (201810489071)
收稿时间: 2018-05-09; 修改时间: 2018-06-04; 采用时间: 2018-06-11; csa 在线出版时间: 2018-12-03

技术, 开发了一个高性能的查询模块的搜索系统. Marin[5]通过对网页中的信进行抽取过滤后, 通过构建垂直搜索工具, 帮助用户查找某个特定领域的相关信息. 近年来, 垂直搜索技术应用相结合在国内也获得了较快的发展. 刘全志等人[6]利用开源的 Heritrix 和 Jsoup 设计了一个网络商品信息抽取系统, 实现了 Web 信息的抽取、存储、检索; 张亚凤[7]通过扩展 Hertrix 框架并添加自定义词库实现了体育用品信息类的垂直搜索引擎, 其添加自定义词库的方法, 对本文扩展完整性词库的研究起到了借鉴作用; 吴洁明等[8]在对 Hertrix 拓展优化的基础上, 通过倒排索引提高了搜索的效率. 张敏等[9]基于 Heritrix 框架实现了通过 IP 地址限制爬虫只抓取某一地区主机上的网页, 采用拓展 Heritrix 爬虫的 Post-processing chain 进行限定性采集, 有较强的实用性. 吴伟等[10]运用 ELFHash 算法对 Heritrix 进行多线程的优化, 增加爬取线程数, 提高了网页抓取的速度. 上述研究总的来说, 对爬虫的抓取效率、正文的抽取精度和搜索精度方面起到一定提升作用, 但也存在有待进一步完善的问题: (1) 信息获取阶段, 多数研究选取 ELFHash 作为 Heritrix 的散列函数, 但我们在实际应用中发现, ELFHash 函数的散列效果并不是很好, 且这些研究大多没有实现增量抓取, 导致大量 URL 的重复抓取, 降低了抓取的效率; (2) 结构化信息处理阶段, 这些研究主要通过对不同网站手工配置不同模版的方式进行正文抽取, 没有实现网站自动化抽取; (3) 建立索引阶段, 一般都是利用基础词库进行分词建立索引, 所提供的信息检索服务精度有待进一步提高; (4) 大多研究只是对构建搜索引擎的某一个或几个过程进行优化, 而非对其各阶段进行整体优化、一体化的实现垂直搜索引擎的自动化构建.

针对上述问题, 本文针对搜索引擎构建的三个阶段 (信息获取阶段、结构化信息抽取阶段和信息存储检索阶段) 进行研究与优化, 并将各过程进行有效连接, 实现了垂直搜索引擎的自动化构建.

1 自动化垂直搜索引擎构建方法

自动化垂直搜索引擎构建流程主要包含四步, 如图 1 所示.

1) 优化爬虫, 优化抓取任务队列分配策略, 采用增量爬取, 提高爬取效率.

2) 使用经过优化后的文本标签路径比算法抽取网页正文, 标题和时间按其分布规律进行抽取, 最后将所有抽取出的部分按字段存储到数据库中.

3) 使用优化后的 C-value 方法抽取油气资源新闻中的完整性词, 建立油气行业专业词库; 使用专业词库进行分词, 创建索引, 提高搜索的精度.

4) 实时获取最新入库文档、抽取领域词、更新词库、实现 Solr 的近实时搜索功能来不断更新索引, 为用户提供最新的搜索服务.

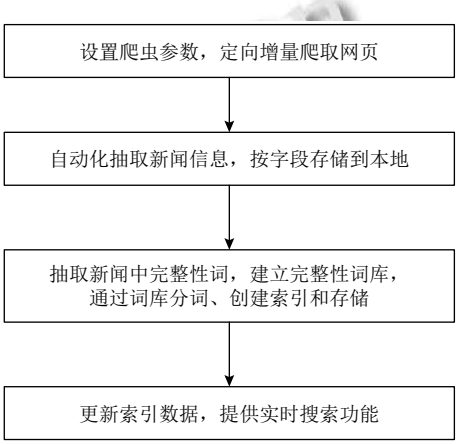

图 1 自动化的垂直搜索引擎构建流程

2 关键问题及解决方案

决定垂直搜索引擎自动化效能的几个关键性问题包括: 1) 爬虫是否高效; 2) 爬取信息能否自动存储在数据库; 3) 是否能建立高效的索引, 为用户提供准确的信息检索服务. 本文对这几方面展开了研究如下:

2.1 Heritrix 爬虫优化

爬取模块使用开源爬虫 Heritrix, 并对其抓取任务的 URL 队列分配策略进行优化, 采用增量爬取方式, 提高爬取效率.

2.1.1 抓取任务中定制 URL 队列分配策略

默认情况下, Heritrix 使用域名分配策略 (Hostname Queue Assignment Policy) 将爬取的 URL 分配到不同的队列抓取. 该策略的特点是: 根据域名作为分组依据, 即同一域名下的 URL 将被分配进同一个抓取队列. 这样在抓取时会单线程抓取. 所以需要将要爬取的多个 URL 分散至多个队列中并行处理, 以提高爬取速度. 大部分解决该问题的方法都是使用 ELFHash 散列函数, 但通过实际测试, 发现效果并不理想[11]. ELFHash 函数将 URL 的绝对长度作为输入, 并与字符的十进制值结合起来计算 hash 值, 这种方式对长字符

串和短字符串都有效, 能够比较均匀地把字符串分布在散列表中. 但采用该函数没有考虑负载均衡, 当种子 URL 较少时, 使用该函数会使得多个线程一起去抓取少量种子, 造成线程阻塞, 无法抽取出新的 URL. 实践中发现, 使用该函数, 爬虫经常会在散列 30 个 DNS 就自动结束.

本文采用的是 Hflp 散列函数[11]. Hflp 函数将不定长的 URL 地址分割成位数相同的几部分, 然后取这几部分的叠加之和 (去除进位) 作为散列地址. 实验证明, 使用 Hflp 散列函数可以提高 Heritrix 的抓取效率, 实现负载均衡.

2.1.2 增量爬虫

增量式爬虫是指对已下载网页进行增量式更新, 只抓取新产生的网页内容, 提高抓取速度[12]. 在互联网信息快速增长的今天, 增量式网络爬虫能够显著的提高网络信息获取效率, 这是因为增量式爬虫具有以下优点:

(1) 因只爬取新产生的或者已经发生变化网页, 对以前曾经爬取过的网页不再抓取, 可有效减少爬取时数据的下载量;

(2) 减小时间和空间上的耗费;

URI (Universal Resource Identifier) 是通用资源标识符的缩写, 唯一标识一个网络资源. URL (Universal Resource Locator) 是统一资源定位符的缩写, URL 是 URI 的子集, 是获取网络资源的唯一标识, 可以通过 URL 来区分不同的网页资源. 通常, 爬虫会将已抓取和待抓取的 URL 分别存放在不同爬取队列中, 并从已抓取的 URL 中发现新的 URL, 经过一系列判断、去重规则过滤后, 将符合规则的 URL 加入待抓取 URL 队列中等待抓取.

Heritrix 爬虫会自动记录每次爬取的 URL, 在爬取结果文件夹 log 文件中的 recover.gz 文件记录了上一次已抓取的 URL 信息. 所以可以对该文件中记录的 URL 不断过滤累加, 每次爬取时, 将这些 URL 写入到已抓取队列, 避免重复信息的二次爬取. 具体改进方法如下:

算法 1. 增量抓取算法

1) 建立 recoverAll.txt 文件, 记录所有已爬取 URL. 当一次爬取完成后, 首先读取 recoverAll 中记录的历史 URL 信息 (首次爬取为空), 并使用指纹算法, 将 URL 压缩后存入 Bloom Filter 中, 生成所有历史 URL 信息的 Bloom Filter;

2) 读取 Heritrix 爬取结果 log 文件夹下的 recover.gz 文件, 该文件记录本次爬取结果, 同样使用指纹算法对有效的 URL 进行压缩处理;

3) 使用步骤 1) 中建立的 Bloom Filter[13]对步骤 2) 中读取的 URL 过滤去重;

4) 将去重后的 URL, 追加至 recoverAll 文件中, 并压缩为 recover.gz 文件;.

5) 将 Heritrix 的配置文件 order.xml 中的<string name="recover-path">属性值更新为新生成的 recover.gz 文件地址, 从而实现对网站长时间监测过程中的增量爬取.

2.2 结构化信息自动抽取方法优化

基于改进的文本标签路径比算法抽取出新闻中的正文, 同时根据新闻标题和时间的分布规律, 实现对标题和时间的精确抽取.

2.2.1 正文的抽取

文本标签路径比算法[14]是一种抽取新闻正文比较有效的方法, 其核心是通过比较正文内容与噪音内容在标签路径和文本内容等特征上的一些显著区别, 抽取出网页中的正文信息.

该算法在处理大多数网页时, 都具有较好的抽取效果. 但对一些短文本新闻会出现抽取错误. 例如在“中国石油新闻中心”网站中, 许多图片新闻只含有一张图片以及对图片的简短描述, 这时算法会将网站“底部版权区”的信息抽取出来作为结果. 错误原因在于, 短新闻的“底部版权区”中的标签数量和“正文”中修饰部分的标签数量相差不大, 但 “底部版权区”中的文字甚至比正文的还多, 而该算法是依据文本长度和标签的数量来计算文本块的分值, 导致最终“底部版权区”的分值高于“正文”部分.

针对这类问题, 本文对该算法进行优化. 我们对中石油新闻中心、中石化新闻网、中海油新闻网三个主要油气信息网站首页共计 306 条新闻的正文部分进行统计, 发现三个网站中的新闻的正文主要可分为三类: (1) 文字类新闻, 这类新闻只包含文字, 不包含图片; (2) 图文交替类新闻, 这类新闻以文字为主, 也包含有部分图片; (3) 图片类新闻, 这类新闻以图片为主, 文字说明很少. 这三类新闻都具有以下特征: (1) 正文中一般都含有标点符号; (2) 正文中一般都不含有超链接; (3)<p>标签和标签一般用于修饰正文中的内容. 对我们监测的其他油气相关网站进行检查, 发现这些网站也具有上述特征.

根据上述特征, 对该算法做出如下改进:

算法 2. 文本标签路径比算法改进

1) 对第一次抽取出的新闻内容进行统计.

2) 确定标点个数是否大于某个阈值 α, 超链接中字符数量是否小于

阈值 β;
3) 若满足上述两个条件, 则认为新闻正文抽取正确, 将第一次抽取的内容作为最终结果. 若上述条件有一个不满足, 则按照新规则进行二次抽取.
4) 在二次抽取中, 统计 DOM 中<p>和标签的个数, 将其作为参数添加至特征值计算公式中, 进行二次计算.
5) 对结果进行排序, 抽取高于阈值的结果, 并按分数高低遍历. 判断: ① 结果中的标点个数是否大于阈值; ② 超链接中的字符数量是否小于阈值. 取出满足①②条件且分数最高的作为最终结果.

2.2.2 标题和时间的抽取

一般来说, 新闻标题在网页中的分布位置, 按优先级排序为: (1) 页面中被<h>标签修饰且和<head>标签中<title>标签内容相似部分; (2) 网页<head>标签中<title>标签的内容; (3) 页面 id 或 class 属性值为 title 的标签的内容.

根据以上规律, 抽取新闻标题方法可分为两种情况处理:

(1) 页面<head>里的<title>标签中有字符;

(2) 页面<head>里的<title>标签中没有字符.

具体抽取算法如下:

算法 3. 正文标题抽取算法

1) 对于情况 (1), 遍历 DOM 到正文元素, 获取<h>标签修饰的字符, 计算字符和<head>里的<title>标签中字符的相似度[15], 若某个<h>标签中的字符和<title>中字符相似, 则将其作为标题. 若页面中所有<h>标签中的字符和<head>里的<title>标签中字符都不相似, 则将<head>里的<title>标签中字符作为标题;
2) 对于情况 (2) 遍历 DOM, 找出 id 或 class 属性的值为 title 的标签, 将第一个找到的字符作为标题内容.

对时间信息的处理方法如下: 由于 Heritrix 会以其爬取的 URL 地址作为路径建立文件夹, 将爬取下来的内容存储在该文件夹中. 例如, http://news.cnpc.com.cn/system/2016/05/09/001591599.shtml 页面下载到本地的存储路径为: 项目根目录/jobs/爬取任务名称/mirror/news.cnpc.com.cn/system/2016/09/001591599.shtml. 而“项目根目录/jobs/爬取任务名称”部分路径可由用户自定义. 根据该特点, 正文的时间抽取算法如下:

算法 4. 正文时间抽取算法.

1) 利用正则表达式将该地址下 mirror 文件夹后面部分内容截取出来;
2) 在截取的部分前面加上爬取网站的网络协议即为爬取的 URL;
3) 在使用正则表达式, 将 URL 中的年月时间信息提取出来, 即为新闻的发布时间.

2.3 基于 C-Value 改进的油气行业词库建立方法

索引的建立是以词为基础的, 分词的准确性是衡量一个垂直搜索引擎搜索精度的重要指标. 基础分词器通常会根据基本词典切分, 往往将有完整意义的词分开, 例如: “中石油”会被分为“中”和“石油”、“国际能源产业”会被分为“国际”、“能源”和“产业”等. 分割开的词无法反映原词本身含义, 通过这样的方式分词建立索引, 只能检索出割裂词的索引信息. 本文通过抽取大量新闻中的具有完整意义的词, 建立完整性词库, 并利用词库进行分词、建立索引, 提高搜索准确度.

2.3.1 C-value 方法

C-value[16]方法是 Frantzi K(1999) 提出的. 该方法基于统计学与语言学规则, 是一种领域独立的多字词语自动抽取方法.

统计部分计算公式如下:

$$C-value(a)=\begin{cases}\log_2|a|*f(a) & a\text{被嵌套}\\ \log_2|a|\left(f(a)-\dfrac{1}{P(Ta)}\sum_{b\in Ta}f(b)\right) & \text{其他}\end{cases} \tag{1}$$

公式 (1) 中 a 表示某候选术语, $|a|$指候选术语 a 的长度, 其值为 a 的字数, $f(a)$ 为 a 出现的词频, b 表示包含 a 的候选术语, $f(b)$ 表示其出现的频率, Ta 表示包含词串 a 的集合. $P(Ta)$ 表示 Ta 的个数. 通过实验发现, 在实际应用中使用 C-value 方法抽取的效果不好. 原因如下:

(1) C-value 的词性过滤规则较为宽松, 虽然一定程度上增加了召回率, 但极大地降低了抽取精度;

(2) C-value 是基于英文环境的, 并不完全适用于中文领域词的抽取.

2.3.2 总结行业词典过滤规则

不同行业的词库可能具有不同的规律, 因此需要根据涉及的领域, 总结出该行业常用词典的一些规律来指导建库. 下面以油气行业为例说明.

搜集腾讯、搜狗等公开的词库、常用分词器的词库以及互联网上常用油气行业词库, 对其进行文本转化, 过滤, 去重后得到油气行业的常用词, 共 30882 条. 对其结构成分进行分析发现:

(1) 油气行业常用词的字数长度变化范围在 1–10 之间. 其中, 长度在 2–5 间的词最多, 有 25 629 条, 占总数的 83%. 长度为 1 的词有 4614 条, 占总数的 15%. 长度 6–10 的只有 639 条, 占 2%. 这与一般来说英语术语多由 2–3 个单词组成, 而汉语术语多由 2–6 个字组成[17]

相符合.

(2) 通过对 30 882 条词语的结构进行统计分析, 总结出长度在 2–5 之间的排名前 20 的词性列表如下:

2.3.3 抽取行业完整性领域词建库的方法

根据得到的词性规则, 采用以下算法抽取行业完整性领域词:

算法 5. 行业完整词抽取算法

1) 使用 ICTCLAS 分词器对爬取下来的新闻进行词性标注, 并利用表 1 中 20 组语言学规则进行过滤, 匹配满足上述 20 组规则的词;
2) 对所有满足规则的词过滤噪声, 去除单字动词、特殊符号等, 将剩下的词作为语言学规则处理后的候选词;
3) 使用公式 (1) 计算语言学规则候选词的 C-value;
4) 使用通用词库, 对计算结果中的常用词进行过滤并排序, 将高于阈值的部分作为最终结果词.

表 1 语言学规则

词性规则	规则实例	词性规则	规则实例
n n	天然气/n 水合物/n	n vn	水力/n 勘探/vn
v n	开采/v 石油/n	n n n	碳/n 水/n 化合物/n
n v	污水/n 处置/v	v g	溶解/v 性/g
vn n	超重/vn 原油/n	b n	无机/b 化学/n
a n	有害/a 气体/n	n g	钨铁/n 石/g
v v	勘探/v 钻孔/v	n v n	燃料/n 脱/v 硫/n
n vn n	水力/n 开采/vn 矿井/n	v n n	凿/v 井/n 技术/n
g n	微/g 污染物/n	v v n	可/v 采/v 矿石/n
n g n	地层/n 弯/g 斜度/n	v g n	冲/v 蚀/g 探测器/n
n v g	岩石/n 掘进/v 机/g	a n n	绿/a 磷/n 铁矿/n

经过多次实验, C-value 的阈值在大于 3.0 抽取效果最好.

3 实验分析

本文以油气行业垂直搜索引擎的构建为例, 实现上述方法, 并对各过程的结果进行记录分析. 实验的硬件环境为: 服务器 CPU 为 Intel Celeron E3300, 双核主频为 2.5 GHz; Heritrix 采用 1.14.4 版本; 爬取最大线程设置为 50, 最大深度设置为 3, 最大爬取时间为 60 min, 其他为 Heritrix1.14.4 默认设置.

3.1 URL 散列效果对比

使用 Heritrix 对中国石油新闻中心 (http://news.cnpc.com.cn/) 进行爬取, 分别采用 ELFHash 函数和 Hflp 函数进行爬取对比. 对 60 分钟内 URL 散列数进行统计, 结果如图 2 所示.

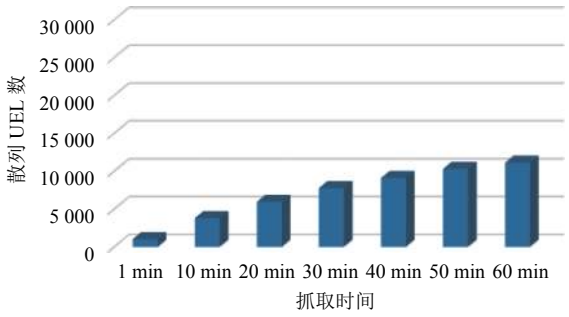

图 2 ELFHash 函数散列 URL 数

从图 2、3 的对比中可以看出, 在 60 分钟内的对各个时间点统计, 可见 Hflp 较 ELFHash 具有更好的散列效果.

3.2 增量爬取

采用 Heritrix 对中国石油新闻中心进行增量抓取和普通 (非增量) 抓取, 监测时间为一周 (网站周末不更新), 每天抓取一次, 抓取结果如图 3 所示.

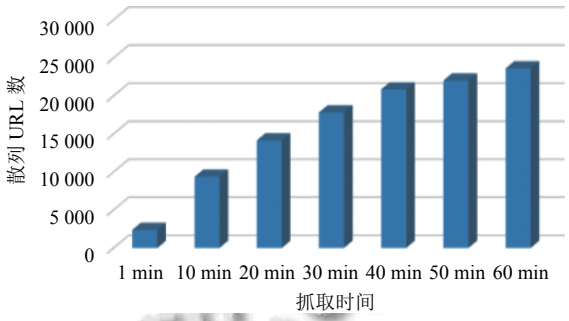

图 3 Hflp 函数散列 URL 数

从图 4、5 对比中可以看出, 在进行普通 (非增量) 爬取时, 由于爬取时间都设置为 1 小时, 所以爬取 URL 数波动范围不大, 维持在 37 000 条左右, 但这些爬取的新闻中包含了网站每天新发布的新闻和前面已爬取过的重复新闻. 而采用增量爬取后, 由于增量的作用每天只爬取新入库的新闻. 虽然爬取时间仍然设置为 1 小时, 但除了第一天爬取了 1 小时外, 随后几天爬取的时间和 URL 都呈现明显下降的趋势, 最后基本趋于稳定. 进一步表明增量抓取能有效提高爬虫的抓取效率.

3.3 结构化信息抽取部分

对中国石油新闻中心、中石化新闻网和中海油集团新闻网爬取的 1000 个网页进行结构化信息抽取, 使用基于文本标签路径比算法和优化后的文本标签路径比算法进行抽取, 并进行效果对比. 采用人工判读的方

式统计数据库中新闻结构化信息抽取正确与否, 结果如表 2 所示.

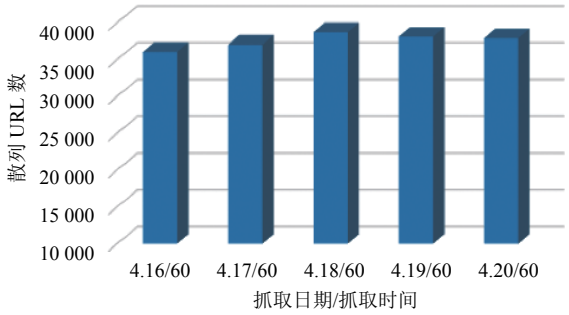

图 4 普通抓取方式抓取的 URL 数 (min)

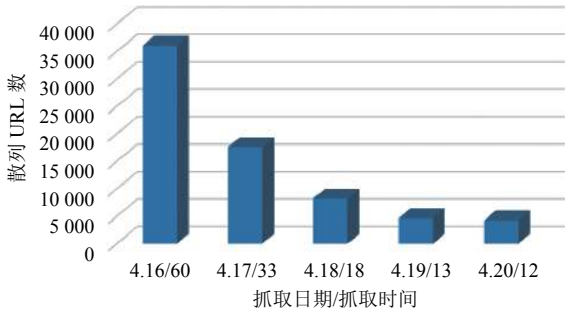

图 5 增量方式抓取 URL 数 (min)

表 2 文本标签路径比算法优化前后结果对比 (单位: %)

监测网站	文本标签路径比算法抽取正确率	文本标签路径比算法优化后抽取正确率
中国石油新闻中心	97.5	99.8
中石化新闻网	84.1	99.6
中海油集团新闻网	95.3	99.1

从表中可以看出, 在对中国石油新闻中心、中石化新闻网和中海油集团新闻网的抽取中, 优化后的文本标签路径比正文抽取算法准确率较优化前有了明显的提升.

3.4 完整词抽取算法

评价抽取效果一般采用精度和召回率两个指标[18]. 本文采用精度、召回率和 *F-score* 三个指标评价完整词的抽取效果. 精度用抽取算法抽取出的完整性词占人工判读标记的完整性词的百分比表示, 而召回率用抽取算法抽取出的正确完整性词数占人工判读标记正确完整性词数的百分比表示. 为综合评价抽取效果, 采用 *F-score* 评价指标, 计算公式如下:

$$F-score=\frac{2\times 精度\times 召回率}{精度+召回率} \quad (2)$$

对中国石油新闻中心爬取的网页清洗后, 选取其中的 50 篇文章, 通过分词、停用词过滤等处理后, 得到 8263 个候选词, 通过精度、召回率和 *F-score* 对传统和优化后 C-Value 方法进行评估, 结果如表 3 所示.

表 3 基于传统 C-value 和行业词典过滤规则改进的 C-Value 方法抽取结果对比 (单位: %)

方法	精度	召回率	*F-score*
传统 C-value 方法	72.93	74.12	73.52
改进 C-Value 方法	77.32	88.63	82.58

从结果可知, 油气行业词典总结出的词性规则较 C-value 的过滤规则更适合提取油气行业完整性词.

3.5 搜索结果比较

本文利用上述垂直搜索引擎自动化构建方法对中国石油新闻中心进行处理, 最后输入关键词进行搜索, 并将搜索结果与未使用专业词库只基于基础词库分词建立索引的搜索结果进行对比.

图 6 是未使用专业词库搜索的结果, 当输入检索词“世界石油大会”后, 检索出的是按照基础词库分词建立索引获取的结果, 检索出的主要是这个词被分词后的子词相关信息, 而包含有完整的“世界石油大会”词的文档并没有优先排在前面, 结果不能让用户满意. 图 7 是使用专业词库后搜索的结果, 这里 “世界石油大会”是作为一个完整意义的词被检索, 并没有被分割为“世界”、“石油”、“大会”三个词分别检索, 检索结果能更好的满足用户对搜索精度的需求.

图 6 未使用专业词库搜索结果

4 结论

垂直搜索引擎技术研究的目的在于让特定搜索领域和搜索需求的用户有更好的用户体验. 本文对垂直

搜索引擎构建中各个阶段的关键问题进行了深入研究, 提出一套自动化构建垂直搜索引擎的方法. 与已有的方法相比, 该方法能够提高爬虫的爬取效率, 能够进行结构化信息的自动化提取, 能够抽取完整性领域词建立油气行业词库, 优化分词方式, 提高搜索的精度. 该方法已经成功应用于国土资源部油气资源战略研究中心的油气资源行业信息网, 为油气及相关从业人员提供快捷、准确和有效的信息检索服务.

图 7 使用专业词库搜索结果

参考文献

1 王泓淼. 网页信息智能采集与个性化服务系统的研究与实现[硕士学位论文]. 天津: 河北工业大学, 2012.

2 De Bra P, Houben GJ, Kornatzky Y, *et al*. Information retrieval in distributed hypertexts. Proceedings of Intelligent Multimedia Information Retrieval Systems and Management. New York, NY, USA. 1994. [doi: 10.1007/0-306-47019-5_5]

3 Chakrabarti S. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann, 2003: 73–74. [doi: 10.1016/j.ipm.2005.06.002]

4 Baeza-Yates R, Gionis A, Junqueira F, *et al*. The impact of caching on search engines. Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Amsterdam, The Netherlands. 2007. 183–190. [doi: 10.1145/1277741.1277775]

5 Castro HMM, Sosa VS, Maganda MAN. Automatic construction of vertical search tools for the deep web. IEEE Latin America Transactions, 2018, 16(2): 574–584. [doi: 10.1109/TLA.2018.8327415]

6 刘全志, 于治楼. 基于 Heritrix 和 Jsoup 的信息抽取系统的设计与实现. 山东师范大学学报 (自然科学版), 2015, 30(2): 16–19. [doi: 10.3969/j.issn.1001-4748.2015.02.005]

7 张亚凤. 垂直搜索引擎中关键技术的研究[硕士学位论文]. 长春: 长春工业大学, 2016.

8 吴洁明, 冀单单, 韩云辉. 基于 Web 的 DCI 垂直搜索引擎的研究与设计. 计算机工程与设计, 2013, 34(4): 1481–1487. [doi: 10.3969/j.issn.1000-7024.2013.04.066]

9 张敏, 孙敏. 基于 Heritrix 限定爬虫的设计与实现. 计算机应用与软件, 2013, 30(4): 33–35, 80. [doi: 10.3969/j.issn.1000-386x.2013.04.010]

10 吴伟, 陈建峡. 基于 Heritrix 的 Web 信息抽取优化与实现. 湖北工业大学学报, 2012, 27(2): 23–26. [doi: 10.3969/j.issn.1003-4684.2012.02.007]

11 李晓明, 凤旺森. 两种对 URL 的散列效果很好的函数. 软件学报, 2004, 15(2): 179–184.

12 孟庆浩, 王晶, 沈奇威. 基于 Heritrix 的增量式爬虫设计与实现. 电信技术, 2014, (9): 97–101. [doi: 10.3969/j.issn.1000-1247.2014.09.021]

13 严华云, 关佶红. Bloom Filter 研究进展. 电信科学, 2010, 26(2): 31–36. [doi: 10.3969/j.issn.1000-0801.2010.02.008]

14 吴共庆, 胡骏, 李莉, 等. 基于标签路径特征融合的在线 Web 新闻内容抽取. 软件学报, 2016, 27(3): 714–735. [doi: 10.13328/j.cnki.jos.004868]

15 姜华, 韩安琪, 王美佳, 等. 基于改进编辑距离的字符串相似度求解算法. 计算机工程, 2014, 40(1): 222–227. [doi: 10.3969/j.issn.1000-3428.2014.01.047]

16 Frantzi K, Ananiadou S, Mima H. Automatic recognition of multi-word terms: The C-value/NC-value method. International Journal on Digital Libraries, 2000, 3(2): 115al lib. [doi: 10.1007/s007999900023]

17 张勇. 中文术语自动抽取相关方法研究[硕士学位论文]. 武汉: 华中师范大学, 2006. [doi: 10.7666/d.y874926]

18 袁劲松, 张小明, 李舟军. 术语自动抽取方法研究综述. 计算机科学, 2015, 42(8): 7–12. [doi: 10.11896/j.issn.1002-137X.2015.8.002]