使用联合链接相似度评估爬取 Web 资源

张乃洲 李石君 余 伟 张 卓

(武汉大学软件工程国家重点实验室 武汉 430072)

(武汉大学计算机学院 武汉 430072)

摘 要 如何从 Web 上获取感兴趣的资源是许多 Web 研究领域重要的研究内容.目前针对特定领域 Web 资源的获取,主要采用聚焦爬行策略.但目前的聚焦爬行技术在同时解决高效率爬行和高质量的爬行结果等方面还存在许多问题.文中提出了一种基于联合链接相似度评估的爬行算法,该算法在评估链接的主题相似度时,联合使用了关于链接主题相似度的直接证据和间接证据.直接证据通过计算链接的锚链文本的主题相似度来获得,而间接证据则是通过一个基于 Q 学习的 Web 链接图增量学习算法获取.该算法首先利用聚焦爬行过程中得到的结果页面,建立起一个 Web 链接图.然后通过在线学习 Web 链接图,获取链接和链接主题相似度之间的映射关系.通过对链接进行多属性特征建模,使得链接评估器能够将当前链接映射到 Web 链接图的链接空间中,从而获得当前链接的近似主题相似度.在 3 个主题域上对该算法进行了实验,结果表明,该算法可以显著提高爬行结果的精度和召回率.

关键词 聚焦爬行;主题相似度;链接评估;Web 链接图;Q 学习

中图法分类号 TP311 **DOI 号**: 10.3724/SP.J.1016.2010.02267

Using a Joint Link Similarity Evaluation Based Method for Crawling the Resources on Web

ZHANG Nai-Zhou LI Shi-Jun YU Wei ZHANG Zhuo

(*State Key Laboratory of Software Engineering*, *Wuhan University*, *Wuhan* 430072)

(*School of Computer*, *Wuhan University*, *Wuhan* 430072)

Abstract For many fields of Web research, how to fetch the interesting resources is crucial. At present, the chief method for obtaining the domain-specific resources on Web is to adopt the strategy of focused crawling. However, for the most current techniques of focused crawling, there are many problems in simultaneously meeting the high efficient crawl and the high quality of crawl results. This paper proposes a joint link similarity evaluation based algorithm. When evaluating the similarity between a link and a specific topic, the algorithm combines the direct evidence with indirect evidence on the topic similarity of the link. The direct evidence can be obtained by computing the topic similarity of the anchor text corresponding to the link. As to the indirect evidence, this paper presents a Q learning based algorithm for incrementally learning Web Link graph. The algorithm firstly builds a Web link graph by exploiting the on-topic Web pages fetched by focused crawler and then gets the map relationship between the link and topic similarity through online learning. Modeling any link as a multi-attribute vector, the system gives the link evaluator the ability to map the current link into the space of the Web link graph and thus

收稿日期:2010-06-11;最终修改稿收到日期:2010-07-02.本课题得到国家自然科学基金(60970018)资助.**张乃洲**,男,1970 年生,博士研究生,主要研究方向为 Web 数据管理、信息集成. E-mail: zhangnz@126.com. **李石君**(通信作者),男,1964 年生,博士,教授,博士生导师,主要研究领域为 Web 数据管理、信息集成. E-mail: shjli@whu.edu.cn. **余 伟**,男,1987 年生,博士研究生,主要研究方向为 Web 数据管理. **张 卓**,男,1978 年生,博士研究生,主要研究方向为 Web 数据管理.

obtains its approximate topic similarity. The experimental results for three specific topics show that the algorithm can significantly improve the precision and the recall of crawl results.

Keywords focused crawling; topic similarity; link evaluation; Web link graph; Q learning

1 引 言

随着 Web 规模日益扩大,Web 已经成为一个巨大的资料库.如何从海量的 Web 资源中发现感兴趣的信息,是目前许多 Web 研究领域(如 Web 信息检索和 Web 信息集成)重点研究的问题.该任务首先要在 Web 上获取目标数据集,然后再对目标数据集做进一步处理,如索引、信息提取、信息集成等.对目标数据集的获取,目前主要采用智能代理程序(也称为 Spider 或 Crawler)来自动收集 Web 页面.Crawler 的主要原理是:从给定的一些种子 URL 出发,采用一定的搜索策略,沿着 Web 超链结构进行爬行,最后达到遍历整个 Web 图的目的.根据任务的性质不同,Crawler 的搜索策略主要分为两种[1-3]:一种是宽度优先搜索(breadth-first search),另一种是最好优先搜索(best-first search).前者典型的应用是通用搜索引擎,其爬行的目标是目标数据集尽可能地完备(即尽可能地遍历整个 Web 图),并且可以通过设置最大爬行深度,来有效地控制爬行的进程.后者典型的应用是垂直搜索引擎,主要用于搜集一些特定领域的信息.由于 Web 的规模呈指数式的增长,并且 Web 内容具有很强的动态性,这使得搜集特定领域信息的任务变得越来越具有挑战性.这种挑战性主要体现在:(1)如何自动发现和定位包含特定领域信息的 Web 页面;(2)如何进行高效率爬行;(3)如何获得高质量的爬行结果.要同时解决以上 3 个方面的问题,需要设计合适的爬行策略和精确的页面分类器.相对于 Web 的规模来说,特定领域的信息具有很强的稀疏性.在这种情况下,采用宽度优先搜索的策略势必造成爬行效率的低下、系统存储空间和网络带宽的浪费.此时,采用最好优先搜索是一种必然的选择.而聚焦爬行(focused crawling)[1-5]是最好优先搜索策略的一种实现.

聚焦爬行的主要目标是:在 Web 上只爬取与主题相关的 Web 页面.理想的聚焦爬虫能够获取最大的相关页面集合而同时遍历最小数量的不相关 Web 文档集合[3],即能够获得最大收益率(harvest rate)[1-2].聚焦爬行实现最好优先搜索策略的基本方法是采用优先队列来存储待爬行的链接①(URL),该数据结构在聚焦爬行框架中也被称为边界管理器(frontier manager)[1-2,4].当聚焦爬虫分析当前 Web 页面时,该页面所包含的链接(针对未被爬行过的 URL)被赋予一个权值,该权值代表了链接将被爬行的优先级.由此可知,聚焦爬虫的关键技术是如何设计高效的链接评估(link evaluation)函数,该函数用于分配链接的爬行优先级.然而在聚焦爬行过程中,最困难的问题恰恰是:在当前链接对应的页面被下载之前,如何判断该链接是否主题相关.此时,常用的方法是对当前链接的特征进行分析,然后做出优先级预测.这些方法包括对链接所对应的锚链文本、URL 字符串、链接环境(Context)进行主题相似度分析.本文称这些方法为链接主题相似度的直接证据分析.但直接证据分析却无法解决普遍存在的延迟收益(delayed benefit)问题.延迟收益[1-2,4-5]问题是指使用直接证据分析当前的链接,得出该链接主题无关.但实际上该链接经过若干个链接之后,又到达了目标页面(主题相关).延迟收益问题会降低系统的爬行收益率.对于链接评估函数的设计,当前研究最多的是基于机器学习的框架[1-5].

虽然目前对聚焦爬行问题已经进行了深入的研究,并且取得了不少的研究成果,也有一些成功的系统被运用于实际.但这些研究和系统在同时解决前面提到的 3 个挑战方面,还存在不少问题,如系统的精度和效率不高、可复用性差等.

本文研究的内容是作者当前一个研究课题的组成部分.该课题的主要任务是以产品搜索为背景,构建一个面向多信息域聚合的产品搜索引擎.该系统的一个重要研究任务是如何构建一个高效的可运用于不同产品域的通用聚焦爬虫.针对当前研究任务的设计目标,本文对聚焦爬行问题进行了系统的研究,提出了一种基于联合链接相似度评估的爬行算法——JLSE(Joint Link Similarity Evaluation).本

① 这里所使用的术语"链接"与超链接、URL 具有相同的语义.在不引起歧义的情况下,本文均用链接来表示超链接或 URL.

文的主要贡献如下：

(1) 提出了一个有效的基于机器学习的链接评估器算法. 该算法联合使用了关于链接主题相似度的直接证据和间接证据. 直接证据是通过计算链接的锚链文本的主题相似度来获取. 间接证据获取的过程为：首先链接学习器根据爬取的结果页面，构造出一个 Web 链接图，然后采用 Q 学习算法学习一个函数 $Q(i)$，该函数将 Web 链接图中的链接 i 映射为该链接的优先级. 所有的链接被建模为一个多维向量，当聚焦爬虫分析当前 Web 页面中的一个链接 u 时，系统会将 u 与 Web 链接图中的每个链接进行比较，然后选择其中相似度最大的链接 v 作为 u 的近似. 此时 u 的爬行优先级可以通过 $Q(v)$ 来计算. 实验表明该算法可以显著提高链接评估器的精度.

(2) 系统具有较高的自动化程度. 除了在系统运行前需要提供一定数量的训练样本之外，系统在运行过程中不需要进行人工干预.

(3) 系统具有很好的可复用性. 当需要将系统运用到不同的产品域时，只需修改系统的配置文件，就能满足对多个产品域的 Web 资源爬取.

本文第 2 节介绍相关的研究工作，包括各种聚焦爬行算法的思想及优缺点；第 3 节给出系统的整体框架以及各主要部件的设计思想和算法，其中重点介绍了链接评估器和链接学习器的设计原理；第 4 节是实验和分析部分；最后是结语以及对未来工作的展望.

2　相关的工作

Chakrabarti 等人[1] 于 1999 年系统地提出了聚焦爬行的基本概念和框架，该框架设计了一个页面分类器，用于评估一个 Web 页面与聚焦主题的相似度. 相似度的大小使用一个概率模型进行计算，并以此控制爬行的方向. 该系统的主要问题是采用假设：一个 Web 页面内所有链接的爬行优先级等于该页面与主题的相似度. 这使得链接爬行优先级分配粒度较粗，没有考虑 Web 页面可能包含多个主题的问题. 随后，Chakrabarti 等人[2] 又对此问题作了进一步改进，如使用基于监督学习的组件——学徒(apprentice)来在线学习链接分类器，该方法引入了反馈机制，以提高链接爬行优先级的分配精度.

在链接分类器的设计方面，Diligenti 等人[3] 提出使用环境图的方法来增强聚焦爬行. 其思想是先根据 k 个种子文档构造 n 层环境图，然后根据环境图构建 $n+1$ 个分类器. 在进行爬行时，对一个给定的 Web 页面分别使用这些分类器将其分类到合适的队列中. 该算法的主要缺点是没有有效利用链接自身的特征. Rennie 和 McCallum[5] 探讨了使用强化学习的思想来训练链接分类器，并用以评估一个给定链接的回报. Assis 等人[6] 提出了一个基于类型感知的聚焦爬行算法，该算法综合利用网页的内容和类型信息来引导爬行过程. Wang 等人[7] 提出了一个基于质心向量的增量式主题爬行算法. 该算法使用 TFIDF22 模型以及 Max、Ave、Sum 3 个启发式规则计算文档特征权重和质心特征权重，并在此基础上构建与根集文档相对应的质心向量，利用它作为前端分类器指导聚焦爬行.

以上提到的文献主要采用以内容分析为主的方法，而该领域另外一个研究方向是基于链接分析的方法. 基于链接分析的聚焦爬行策略借鉴了 PageRank 和 Hits 算法的思想，在聚焦爬行过程中，通过分析 Web 超链结构，对待爬行的 URL 分配适当的优先级. 如 Abiteboul 等人提出了 OPIC 算法[8]，该算法使用 Cash 表示页面重要度，类似 PageRank，通过对 Web 超链结构的分析，对每个页面的 Cash 进行在线动态分配. Guan 等人[9] 改进了 OPIC 算法，提出了 OPIE 算法，该算法在分配 Cash 时，采用了偏置(bias)的方法，而不像 OPIC 算法采用平均分配 Cash 的策略. 这使得 Cash 的分配更倾向于主题相关的 URL. 该方法的优点是结合了内容分析和链接分析，但缺点是计算未下载链接的主题相似度所采用的方法较简单，只是对链接环境主题相似度和链接所在页面的主题相似度进行加权计算. Peng 等人[10] 探讨了聚焦爬行中的隧道穿越(tunneling)问题，提出了两种方法来分别解决灰色隧道和黑色隧道穿越问题. 而针对 Web 上较稀疏的地理位置信息，Ahlers 和 Boll[11] 给出了一个基于贝叶斯分类器的方法来提取 Web 页面和链接图的地理空间属性，从而能够精确发现和索引 Web 上与地理位置信息相关的文档. 而 Yang 等人[12] 提出了一个基于站点层知识(site-level knowledge)的方法来增量爬行 Web 论坛. 该方法通过挖掘 Web 论坛内的站点链接结构来获取站点层的知识. Alam 等人[13] 给出了一个基于分步 PageRank(Fractional PageRank)算法的聚焦爬虫，用于爬行重要的 Web 页面. 该算法的主要思想是：一个 Web 页面的分步 PageRank 值

等于从种子页面到该页面的所有路径的概率总和.

对深网(deep Web)的聚焦爬行方面,Barbosa和Freire[4]给出了一个有效的爬行特定领域的深网表单框架.爬虫在爬行过程中,能够自动学习"有希望"的链接模式,并调整它的爬行策略.

3 系统框架及算法

本节给出基于JLSE算法的系统框架,并对系统各主要部件的功能做详细的讨论.该系统的主要框架如图1所示,系统的主要工作流程如下:

(1)首先,以手工方式构建样本数据库.该数据库中提供了特定主题域的样本页面,样本页面包括正例和反例.

(2)使用样本数据库训练页面分类器.

(3)提供少量种子URL,这些URL是系统爬行的入口,并被添加到边界管理器(frontier manager)中.边界管理器采用优先队列来存储待爬行的链接,每个链接被赋予一定的优先级.种子URL被设置为相同的优先级.

(4)如果边界管理器为空或结果页面数据库中的记录数达到预设的数量,则转入(6);否则,系统从边界管理器中取出优先级最高的URL,并使用页面下载器下载对应的Web页面,然后调用页面分类器对该页面进行分类.如果该页面被分类为反例,则丢弃该页面,返回到(4);否则继续使用结果页面分类器判断该页面是否是目标页面,如果是,将该页面加入结果页面数据库中,返回到(4);否则转入(5).

(5)提取当前Web页面中所包含的所有URL,然后使用链接评估器分配这些URL的爬行优先级,并将URL插入到边界管理器中,然后转入到(4).

(6)系统停止工作.

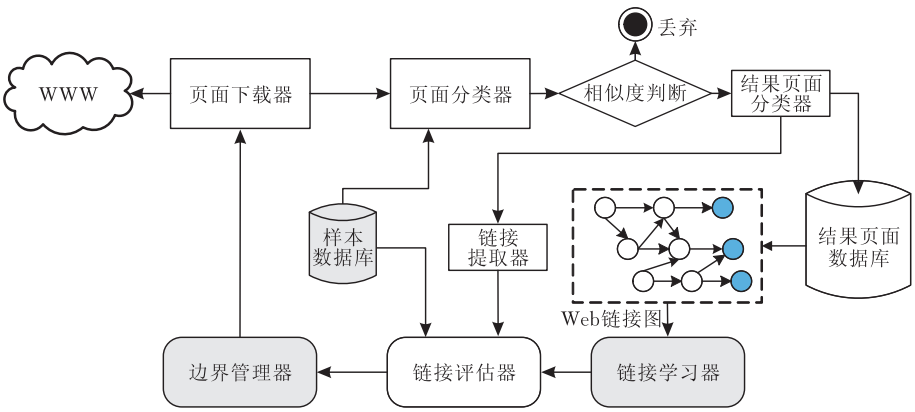

图1 基于联合链接相似度评估的聚焦爬虫架构

3.1 页面分类器和结果页面分类器

页面分类器主要用于判断当前下载的页面是否主题相关.在整个系统中,页面分类器扮演着非常重要的角色.首先它被用于发现主题相关的页面.其次,被用于发现哪些页面可能包含潜在的主题相关链接.对于后者,本文所采取的策略是:只爬取主题相关的页面.为了证明该策略的合理性,先给一些相关的定义.

定义1. 文档定义为集合 $D=\{w|w$ 为文档中出现的词$\}$.

定义2. 主题定义为集合 $T=\{D_1,D_2,\cdots,D_n\}$,其中 D_i 是文档,并且 T 中所有文档描述相同的语义概念.

定义3. 设一个Web页面 u 是一个文档,则 u 与主题 t 的相似度定义为一个函数 $f:S\times\Theta\rightarrow\Re^+$.其中,$S$ 是文档的集合,Θ 为主题的集合,$\Re^+$ 为正实数集.

页面的主题相似度函数 $f(u,t)$ 用来度量页面 u 与主题 t 在内容上相似的程度.本文的页面主题相似度计算采用了概率模型的方法(参见本节式(1)).

定义4. 对页面 u 和 v,如果 $|f(u,t)-f(v,t)|<\Delta$,则称 u 和 v 内容相似.其中,Δ 为一阈值.

假设1. 如果页面 u 到页面 v 存在链接关系,则 u 与 v 内容相似.

下面说明假设1的合理性.由超级链接的定义①和Web页面设计常识可知,一般说来,当页面作者将超级链接添加到Web页面,其意图是:(1)对当前页面的内容或其中的一些概念做进一步补充说明;(2)表明该页面内容的引用来源;(3)列出与当前页面内容相关的页面链接等.但无论何种情况,都能说明:当前页面和链接所对应的页面之间一定是

① http://en.wikipedia.org/wiki/Hyperlink

关于同一主题的，且对该主题，两个页面的主题相似度相差不大，即在内容上是相似的(定义 4)，否则无法解释页面作者添加该链接的动机.

但在现实条件下，却存在其它一些情形，使得假设 1 不能完全成立. 如目前的 Web 页面往往出于商业目的或其它动机，在网页模版中包含了许多与主题无关的链接，如导航栏、广告链接、友情链接等. 在这种情况下，对于一个主题相关的页面来说，如果它包含了这些“噪音”链接，那么显然假设 1 对这些链接是不成立的. 但此时一个基本事实是：这些链接的存在不会影响该页面中其它链接的主题相似性. 又如目前许多 Web 页面往往包含多个主题概念块(页面块)，它们分属多个主题. 在这种情况下，针对某个主题，页面中主题相关的链接可能只包含在某个或几个主题相关的主题概念块中. 因此，在上述条件下，要获取一个主题相关页面中所有主题相关的链接，需要在假设 1 的基础上，结合对链接本身的相关特征信息分析，来进一步判断链接的主题相关性. 这也是本文的基本思路.

从假设 1 可知，如果页面 u 主题无关，则 u 中包含的任意链接 i 对应的页面 v 主题无关(因为 u 与 v 内容相似). 即本文采用的只爬取主题相关页面的策略是合理的.

分类器的设计主要采用文本分类技术，如支持向量机(SVM)、朴素贝叶斯(NB)、决策树(DT)、人工神经网络(ANN)等方法. 本文涉及的页面分类主要是两类分类问题(主题相似与不相似)，页面分类器采用了朴素贝叶斯分类器(Naïve Bayes Classifier)，其计算公式为

$$P(c_i|d_j)\propto\frac{1}{P(d_j)}P(c_i)\prod_{w_k\in d_j}P(w_k|c_i)\qquad(1)$$

其中，d_j 为待分类的 Web 文档，c_i 表示类别. d_j 的特征表示为向量$\langle w_1,w_2,\cdots,w_n\rangle$，$w_i$ 为文档中出现的词. 为了减少计算复杂度，在计算概率时采用了一元模型的假设，即不考虑词在文档中的顺序关系，词与词在文档中的出现是相互独立的. $P(d_i)$ 在计算过程中是一个常数，而 $P(w_k|c_i)$ 的计算方法为

$$P(w_k|c_i)=\frac{1+\sum_{d_s\in c_i}N(w_k,d_s)}{|V|+\sum_{w_t\in V}\sum_{d_s\in c_i}N(w_t,d_s)}\qquad(2)$$

其中，$N(w_k,d_s)$ 表示词 w_k 在文档 d_s 中出现的次数. V 表示 c_i 类的词汇表. 式(2)使用了平滑技术，避免了词分布的稀疏性带来的零概率的问题.

本文对目标页面的判断使用了一个基于支持向量机(SVM)的结果页面分类器. 本系统的目标是获取产品数据页面，以便提取其中的产品信息. 为了使结果页面适合充当产品数据的信息提取页面，我们将产品数据页面分为两类：链接页(Link Page)和细节页(Detail Page). 链接页的特点是：页面具有较多的链接而文字较少，适合 Crawler 进行爬取. 而细节页以文本为主，适合产品信息的提取. 因此本系统的结果页面应当属于细节页. 但由于目前的细节页出于商业目的，在网页模版中往往包含了许多与主题无关的链接，如导航栏、广告链接等，因此仅靠简单统计页面的链接数和文本数并不能做出准确判断. 根据观察，我们使用三元组$\langle TextDegree, LinkDegree, Aggregation\rangle$来表示这两种类型的页面特征，每个分量的含义如下.

(1) *TextDegree*(页面文本度).

$TextDegree=\log_2(plaintexts/k)$. 其中，*plainTexts* 表示整个文档中去除链接文本后的文本数(单词数)，k 为一经验常数. *TextDegree* 越大，该页面属于细节页的可能性越大.

(2) *LinkDegree*(页面链接度).

$LinkDegree=\log_2(linknums/TotalWords)$. *TotalWords* 代表整个文档中的文本数(单词数)，*linknums* 表示该文档中的链接数目. *LinkDegree* 越大，该页面属于 Link Page 的可能性越大.

(3) *Aggregation*(文本聚集度).

一般 Detail Page 的显著特征是文本在页面中以文本块(block)的方式出现. 因此，*Aggregation* 越大，该页面属于 Detail Page 的可能性越大. 该特征主要是通过计算整个文档中文本块(block)的个数和大小获得的.

基于 SVM 的结果页面分类器的工作过程如下：

(1) 构造结果页面训练样本集. 该训练样本集只包含正例和反例. 关于结果页面训练样本集的具体情况可以参看 4.1 节的数据集部分.

(2) 对每一个训练样本，使用前面提到的三元组页面特征向量进行特征表示，并加上类别标注(如正例标注为 1.0，而反例标注为 0.0).

(3) 在训练样本集上训练 SVM 分类器，最后得到训练模型. 系统采用高斯径向核函数作为空间变换函数.

(4) 对于待分类的页面，先进行如同(2)的特征表示，然后使用训练成功的模型进行分类.

3.2 链接评估器

链接评估器是整个系统的核心，它用于预测当前页面中的链接与主题的相似度. 由于此时链接对应的页面还未被下载，无法直接计算其页面的主题相似度，所以对链接与主题相似度的分析通常采用基于链接特征的分析方法[2,14]. 常用的链接特征有锚链文本(Anchor Text)、URL 字符串、链接环境(Context)以及当前页面. 链接环境是指包围在当前链接周围的页面元素，如文本、链接等. 在这些特征中，当前页面的主题相似度常常被采用. 为了说明这种方法的有效性，给出定理 1.

定理 1. 对任意页面 u，可以用 u 的主题相似度来估计 u 中任意链接 i 对应的页面 v 的主题相似度.

证明. 对一特定主题 t，设 u 和 v 的主题相似度分别用 $f(u,t)$、$f(v,t)$ 来表示. 由假设 1 可知，u 与 v 在内容上相似，即 $f(u,t)=f(v,t)+\Delta$，其中，Δ 为一较小正实数. 即可以用 u 的主题相似度来估计 v 的主题相似度. 证毕.

定理 1 表述了这样一个思想：在 Web 上，两个有链接关系的页面具有相似主题的概率远远大于两个随机选择的页面. Davison[14] 的研究也证明了定理 1 的正确性. 这为预测 Web 页面中链接的主题相似度提供了一个基本的方法. 实际上，许多聚焦爬行算法都使用了定理 1 所阐述的思想[1-3].

其它的特征中，由于 URL 字符串包含的信息较少，所以在使用上比较困难. 链接环境有一定的分类作用，但如果链接环境包含的主题与链接主题不一致，则会产生偏差. 而锚链文本具有比较大的分类价值[14].

如引言中所述，在实际应用中，如果只使用上述的直接证据分析，将无法解决延迟收益(delayed benefit)问题. 因此，本文在设计链接评估器时，采用了联合评估的策略，链接主题相似度的计算采用如下公式：

$$sim(u,t)=\max\{sim_{\text{anchor}}(u,t),sim_{\text{predicted}}(u,t)\} \tag{3}$$

其中，$sim(u,t)$为当前页面中，某一链接 u 与主题 t 的相似度. $sim_{\text{anchor}}(u,t)$为采用锚链文本为度量所得到的主题相似度，它提供了主题相似度的直接证据. $sim_{\text{predicted}}(u,t)$为 u 的预测主题相似度，它提供了主题相似度的间接证据. 计算 $sim_{\text{anchor}}(u,t)$的算法描述如下.

算法 1. GTSAT(Get the Topic Similarity of Anchor Text)算法.

输入：当前链接 u 的锚链文本，样本数据库

输出：当前链接 u 的锚链文本主题相似度 $sim_{\text{anchor}}(u,t)$

Begin

1. 从样本数据库中获取每一样本页面在父页面中所对应的锚链文本(手工构建样本数据库时，已经获取了该信息).

2. 对该锚链文本进行分词，去除停用词等处理，然后获取所有的词汇，并保存为一个正例文档或反例文档.

3. 对当前链接 u 的锚链文本进行分词，去除停用词等处理，获得所有词汇.

4. 使用式(1)计算当前链接 u 是否属于正例(主题相似)，如果是，将式(1)计算的结果作为 $sim_{\text{anchor}}(u,t)$；如果属于反例，则将一较小的实数(如 0.01)赋给 $sim_{\text{anchor}}(u,t)$.

5. 返回 $sim_{\text{anchor}}(u,t)$.

End.

而 $sim_{\text{predicted}}(u,t)$的计算需要借助链接学习器学习到的 Web 链接图 G_w 的知识. G_w 提供了成功的爬行路径信息，本文使用该信息来预测当前链接的主题相似度. 下面先给出 G_w 的定义.

定义 5. 一个 Web 链接图 G_w 是一个有向图 $\langle V,E\rangle$，其中 V 为顶点的集合，$V=\{v|$从结果页面数据库中的记录出发，反向构造出到种子 URL 的路径，v 为路径上的一个节点$\}$. E 为有向边的集合，$E=\{\langle u,v\rangle|u,v\in V$，且 u 到 v 存在链接关系$\}$. G_w 中对应于结果页面数据库记录的顶点称为叶节点.

3.2.1 URL 建模

为了计算 $sim_{\text{predicted}}(u,t)$，本文将任意链接 u 建模为一个三元组：

$$\langle AnchorText,\ Url,\ Context\rangle \tag{4}$$

式中，$AnchorText$ 为链接 u 的锚链文本，Url 为链接 u 对应的 URL 字符串. $Context$ 表示链接 u 在当前页面中的环境. $Context$ 可以通过设置一个窗口尺寸 δ(与 u 的距离)来获取. 文本使用 DOM 树来表示一个 Web 页面，然后获取所有与链接 u 的距离小于等于 δ 的 DOM 树节点的文本. 根据相关文献中的数据[2]和我们的实验分析，在实际应用中，δ 设置为 4～6 为宜. 若 δ 设置得过小(如小于 4)，那么获取的 $Context$ 文本可能会太少，导致无法获取足够的 $Context$ 信息. 而如果 δ 设置得过大(如大于 6)，又可能会引入与链接主题不一致的 $Context$ 信息.

定义 6. 给定一个链接 u，按式(4)中的第 i 个特征，提取其中相应的文本，并对其进行分词、去除停用词等处理后，可以得到一个文档，称该文档

为 $d_u^{(i)}$.

定义 7.　$D_W^{(i)}$ 是一个集合 $S=\{d_v^{(i)} \mid v$ 是 G_w 中的顶点$\}$.

3.2.2　$sim_{\text{predicted}}(u,t)$的计算

对于当前待计算优先级的链接 u，先分别计算出 $d_u^{(1)}$、$d_u^{(2)}$ 和 $d_u^{(3)}$. 然后采用 Okapi BM25① 文档检索模型来分别对这 3 个特征进行计算：

$$sim(d_u^{(i)},d_v^{(i)})=\sum_{t\in d_u^{(i)}} idf(t)\frac{f(t,d_v^{(i)})(k_1+1)}{k_1\left(b\dfrac{ld_v^{(i)}}{avgld_v^{(i)}}+1-b\right)+f(t,d_v^{(i)})} \quad (5)$$

式中，$d_u^{(i)}$ 的含义见定义 6. $d_v^{(i)}$ 的含义与 $d_u^{(i)}$ 相同，不同的是 v 只能取 G_w 中的顶点. $f(t,d_v^{(i)})$表示词 t 在 $d_v^{(i)}$ 出现的次数. $ld_v^{(i)}$ 是 $d_v^{(i)}$ 的长度，而 $avgld_v^{(i)}$ 是 $D_W^{(i)}$ 中所有文档的平均长度. 对于一个确定的 G_w 来说，该值是一个常数. k_1 和 b 是常数，通常取 $k_1=2$，$b=0.75$. $idf(t)$表示转置文档频率，采用如下公式计算：

$$idf(t)=\log\frac{N-df(t)+0.5}{df(t)+0.5} \quad (6)$$

其中，N 为 $D_W^{(i)}$ 中文档的个数，$df(t)$表示在 $D_W^{(i)}$ 中，词 t 至少出现过一次的文档的数量.

对链接 u 按照式(5)进行计算后，可以得到一个相似度矩阵：

$$\boldsymbol{A}=(\boldsymbol{a}_1,\boldsymbol{a}_2,\boldsymbol{a}_3) \quad (7)$$

其中 $\boldsymbol{a}_i=(sim(d_u^{(i)},d_{v_1}^{(i)}),\ sim(d_u^{(i)},d_{v_2}^{(i)})\cdots sim(d_u^{(i)},d_{v_n}^{(i)}))^{\mathrm{T}}$，$n=|D_W^{(i)}|$. 这里，需要对 $\boldsymbol{a}_i$ 进行规范化处理.

给定一个权重向量 $\boldsymbol{w}=(w_1,w_2,w_3)^{\mathrm{T}}$，$w_1+w_2+w_3=1$，最后可以得到一个关于链接 u 的评分向量：

$$\boldsymbol{score}=(s_1,s_2,\cdots,s_n)^{\mathrm{T}}=\boldsymbol{Aw} \quad (8)$$

式中，s_i 表示链接 u 与 G_w 中的顶点 v_i 的加权平均相似度. 取 ***score*** 中的最大分量值 s_j，设其对应于 G_w 中的顶点 v_j，则将 v_j 作为链接 u 在 G_w 中的最相似节点. 通过反复进行多组实验，最终权重向量 $\boldsymbol{w}$ 设为 $(0.5,0.2,0.3)^{\mathrm{T}}$. 从 $\boldsymbol{w}$ 的分配可以看出，在分类作用方面，锚链文本最大，链接环境次之，URL 字符串最小.

为了说明 $sim_{\text{predicted}}(u,t)$的计算，这里先对链接学习器的主要工作过程做一简要介绍. 链接评估器工作在两个不同的阶段. 第 1 阶段，链接评估器主要使用样本数据库中的数据，采用 $sim_{\text{anchor}}(u,t)$进行链接主题相似度的评估. 此时式(3)中，$sim_{\text{predicted}}(u,t)=0$，因为当前结果页面数据库的数据还没有达到给定的数量，链接学习器还未开始工作. 第 2 阶段，链接学习器从结果页面数据库中构建 Web 链接图 G_w，然后使用 Q 学习算法学习系统的最优策略 π^*，即给出一个函数 $Q(v)$，该函数将 G_w 中的一个链接 v 映射为 v 的爬行优先级. 在前面的描述中，我们已经将链接 u 映射到了 G_w 空间中的节点 v_j，它与 v_j 的相似度为 s_j. 此时链接 u 的 $sim_{\text{predicted}}(u,t)$采用下面的公式计算：

$$sim_{\text{predicted}}(u,t)=s_j * Q(v_j) \quad (9)$$

关于链接学习器将在下节做详细的讨论. 重新考虑式(3)，式(3)取 $sim_{\text{anchor}}(u,t)$ 和 $sim_{\text{predicted}}(u,t)$ 之中的最大值作为链接 u 的主题相似度. 但目前还存在一个问题：对于两个分属不同页面的链接 u_1 和 u_2，如果按照式(3)，它们取得了相同的主题相似度，那么如何区分 u_1 和 u_2 的优先级的大小？这里，给出推论 1.

推论 1.　如果页面 u 的主题相似度大于页面 v 的主题相似度，则 u 中所包含的链接的主题相似度大于 v 中所包含的链接的主题相似度.

证明.　对一特定主题 t，设 u 和 v 的主题相似度分别用 $f(u,t)$、$f(v,t)$来表示. 任取 u 中的一个链接 i 和 v 中一个链接 j，由定理 1 可知，链接 i 和链接 j 的主题相似度可分别用 $f(u,t)+\Delta$ 和 $f(v,t)+\Delta$ 来表示，其中，Δ 为一较小正实数. 已知 $f(u,t)>f(v,t)$，所以链接 i 的主题相似度大于链接 j 的主题相似度.　　证毕.

推论 1 表明：计算页面内链接的主题相似度时应当考虑当前页面的主题相似度. 由此，本文采用如下公式来计算最终一个页面 k 中的所有链接的爬行优先级：

$$priority(i)=\frac{sim(i,t)}{\sqrt{\sum_{u\in O_k} sim(u,t)^2}}\times P(c_j\mid d_k) \quad (10)$$

其中，$sim(i,t)$为采用式(3)计算所得到的当前页面 k 中，链接 i 的主题相似度. $O_k=\{v \mid v$ 为页面 k 中的链接$\}$. $P(c_j\mid d_k)$为使用式(1)计算所得到的当前页面 k 的主题相似度.

式(10)的思想与 Guan 等人采用的策略[9]类似：一方面考虑不同的页面主题相似度对所包含链接的主题相似度的影响；另一方面，也考虑到在同一页面中，优先级的分配应该倾向于主题相似度大的

① http://en.wikipedia.org/wiki/Probabilistic_relevance_model_(BM25)上可查阅有关 BM25 的相关知识. http://nlp.uned.es/~jperezi/Lucene-BM25/提供了一个基于 Lucene的 java 包.

链接.另外,该公式也反映了链接结构对相似度的影响,即当前页面中包含的链接(链出)越多,则各链接分配到优先级越低(类似 PageRank).该策略使得爬虫在页面主题相似度相同的情况下,优先爬取链出较少的页面.以上所讨论的链接优先级计算的算法描述如下.

算法 2. GPAOGP(Get the Priority of All Outlinks on a Given Page)算法.

输入:当前页面 k,样本数据库,G_w

输出:当前页面 k 中所有链接的优先级

Begin

1. 使用式(1)计算当前页面 k 的主题相似度 $P(c_j|d_k)$.如果当前页面 k 是反例,则丢弃该页面,转到步 10;否则使用结果页面分类器,判断当前页面是否是目标页面.如果是,则将其加入结果页面数据库,转到步 10;否则使用链接提取器,提取当前页面 k 中包含的所有链接.

2. 取当前页面 k 中包含的任意一个未被计算的链接 u,使用算法 1 计算 $sim_{\text{anchor}}(u,t)$.

3. 使用式(4)对 u 建模,生成 $d_u^{(1)}$、$d_u^{(2)}$ 和 $d_u^{(3)}$.

4. 使用式(4)对 G_w 中任意顶点 v 建模,生成 $d_v^{(1)}$、$d_v^{(2)}$ 和 $d_v^{(3)}$.

5. 使用式(5)计算链接 u 和 G_w 中的任意顶点 v 的相似度,得到式(7)所描述的相似度矩阵 $\boldsymbol{A}$.

6. 使用式(8)计算关于链接 u 的评分向量 ***score***,取 ***score*** 中的最大分量值 s_j.设 s_j 对应 G_w 中的节点为 v_j,则使用式(9)计算链接 u 的 $sim_{\text{predicted}}(u,t)$.

7. 结合步 2 的计算结果,使用式(3)计算链接 u 的 $sim(u,t)$.

8. 如果当前页面 k 中还包含未被计算的链接,则转到步 2;否则,转到步 9.

9. 使用式(10)计算当前页面 k 中任意链接 i 的主题相似度 $priority(i)$.

10. 返回当前页面 k 中所有链接的优先级.

End.

3.3 链接学习器

该节我们将详细介绍链接学习器的工作原理.在上节关于链接评估器的讨论中,我们知道链接评估器在计算一个给定链接的主题相似度时,需要用到 $Q(v)$,而 $Q(v)$ 正是链接学习器的输出.链接学习器的主要工作集中在 G_w 的生成以及对 G_w 的学习,其中核心算法是基于 Q 学习的 Web 链接图学习算法.

3.3.1 Web 链接图的生成

Web 链接图的含义可以参见定义 5.使用 Web 链接图的动机是利用聚焦爬虫在爬行过程中产生的目标页面,给系统提供一个反馈,以提高链接评估器的精度.爬虫产生的目标页面提供了成功的爬行路径信息,这种信息是可以被系统所利用的,并可以用于解决延迟收益(Delayed Benefit)问题.

定义 8. 当结果页面数据库中的记录数 N 达到一个给定的常数 cf 时,系统启动链接学习器对 G_w 进行构建和学习.之后,每当 N 达到 cf 的倍数时,系统会再次启动链接学习器对 G_w 进行重新构建和学习.这里,称常数 cf 为增量学习因子.

由定义 8 可知,对 Web 链接图,系统采用了增量构建和学习的策略.增量学习因子 cf 的取值通过实验确定,在后续的实验部分,我们将对增量学习因子的实验情况做详细讨论.

图 2 是一个 G_w 的例子,其中深色节点代表叶节点(结果页面).G_w 的生成并不复杂.在系统的数据库设计中,有两个表可以用于 G_w 的生成.一个是 $LinkTable\langle ID,url,parentID,depth,anchorText\rangle$,另一个是 $ResultTable\langle ID,urlID\rangle$.其中,$LinkTable$ 用于保存链接结构的信息,而 $ResultTable$ 用于保存结果页面.$ResultTable$ 中的 $urlID$ 为 $LinkTable$ 中的 ID.这样,便可以从 $ResultTable$ 中的记录出发,利用其 $urlID$,在 $LinkTable$ 中找到与之对应的 $parentID$,然后再在 $LinkTable$ 中找到该 $parentID$ 的 $parentID$,依此类推.由此便可以反向构造出所有的从结果页面到种子 URL 的路径.

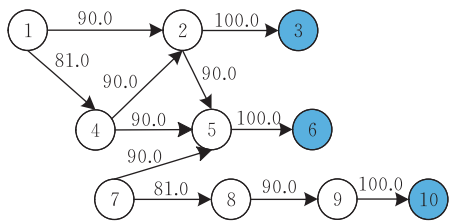

图 2 Web 链接图实例

3.3.2 Q 学习算法

构造 Web 链接图之后,应该如何有效利用成功的爬行路径信息,来进行链接相似度预测呢?基于强化学习(reinforcement learning)的思想[5,15],为利用 Web 链接图提供了一个有效的途径.

强化学习使用一个能够感知环境的代理(Agent),通过环境的反馈,学习最优的动作.强化学习本质上属于马尔科夫决策过程 MDP(Markov Decision Process),其基本框架如下:

(1) 基本定义

状态集合 $S=\{s_0,s_1,\cdots,s_n\}$;

动作集合 $A=\{a_0,a_1,\cdots,a_n\}$;

状态转换函数:$\delta:S\times A\rightarrow S$;

立即回报函数:$r:S\times A\rightarrow\Re$.其中,$\Re$ 为实数集.

(2) 基本原理

在 t 时刻，当 Agent 在一个给定的状态 s_t，选择了一个动作 a_t 后，环境会立即给出当前的回报 $r_t=r(s_t,a_t)$，并迁移到下一个状态 $s_{t+1}=\delta(s_t,a_t)$. Agent的基本任务是学习一个策略：$\pi: S\rightarrow A$，并且使得该策略最优. 为了描述最优策略，给出关于累积回报的定义：

$$V^{\pi}(s_t)=\sum_{i=0}^{\infty}\gamma^i r_{t+i} \tag{11}$$

其中 γ 为折扣率，$0\leq\gamma<1$. 式(11)表示 Agent 从开始状态 s_t 使用策略 π 进行动作选择，在经过若干步之后获得的回报之和. 因此，最优策略 π^* 应该为：Agent 从任意状态 s 出发，该策略都能使累积回报达到最大. 因此 π^* 可以表示为

$$\begin{aligned}\pi^*&=\arg\max_{\pi}V^{\pi}(s)\\&=\arg\max_{a}[r(s,a)+\gamma V^{\pi^*}(\delta(s,a))]\end{aligned} \tag{12}$$

(3) Q 学习

式(12)却无法直接使用. 在实际应用中，是通过引入 Q 评估函数来进行最优策略的学习. Q 函数定义为

$$Q(s,a)=r(s,a)+\gamma V^{\pi^*}(\delta(s,a)) \tag{13}$$

又因为 Q 和 V^{π^*} 存在如下关系：

$$V^{\pi^*}(s)=\max_{a'}Q(s,a') \tag{14}$$

所以式(13)可以改写为

$$Q(s,a)=r(s,a)+\gamma\max_{a'}Q(\delta(s,a),a') \tag{15}$$

式(15)使用了递归定义，所以可以采用动态规划算法来进行有效地计算. 而最优策略 π^* 为

$$\pi^*(s)=\arg\max_{a}Q(s,a) \tag{16}$$

对 $\forall\langle s,a\rangle$(状态-动作对)，可用式(15)进行若干次迭代计算. 设第 n 次迭代得到的当前 $Q(s,a)$ 的近似值表示为 $\hat{Q}_n(s,a)$，则当满足特定的条件时，随着 n 的增加，$\hat{Q}_n(s,a)$ 会收敛到全局最优值 $Q(s,a)$. 该特定条件为：(1) 系统可以建模为一个确定性的 MDP；(2) 立即回报值有界；(3) 每个状态-动作对都被无限频繁地访问. 收敛性证明的关键是最大误差项的更新误差按因子 γ 减小. Mitchell[15] 给出了收敛性的严格证明.

3.3.3　学习 Web 链接图

基于 Q 学习算法的思想，我们设计了一个 Web 链接图的学习算法. 首先定义如下的基本概念：

(1) 状态集合 $S=\{u_1,u_2,\cdots,u_n\}$. 在 Web 链接图中，顶点代表一个状态.

(2) 终点状态集合 $T=\{u_1,u_2,\cdots,u_s\}$. 在 Web 链接图中，一个叶节点代表一个终点状态.

(3) 动作集合 $A=\{e_1,e_2,\cdots,e_m\}$. 在 Web 链接图中，一个有向边代表一个动作，有向边 e_k 可以表示为 $\langle u_i,u_j\rangle$.

(4) 状态转换函数：在 Web 链接图中，如果顶点 u_i 到顶点 u_j 存在一个有向边 $\langle u_i,u_j\rangle$(动作)，则输出状态 u_j.

(5) 立即回报函数：在 Web 链接图中，如果一个有向边 $\langle u_i,u_j\rangle$ 的端点 $u_j\in T$，则给出立即回报 rd(非零实常数)；否则给出立即回报 0.0.

基于以上的定义，对式(15)进行改写，得到 Web 链接图中任意顶点的 $Q(u_i,\langle u_i,u_j\rangle)$ 的计算公式：

$$\begin{aligned}&Q(u_i,\langle u_i,u_j\rangle)=\\&\quad r(u_i,\langle u_i,u_j\rangle)+\gamma\max_{\langle u_j,u_m\rangle\in E}Q(u_j,\langle u_j,u_m\rangle)\end{aligned} \tag{17}$$

然后计算 $Q(u_i)$：

$$Q(u_i)=\max_{\langle u_i,u_j\rangle\in E}Q(u_i,\langle u_i,u_j\rangle) \tag{18}$$

下面给出基于 Q 学习的 Web 链接图学习算法描述.

算法 3.　QLAWLG(Q Learning Based Algorithm for Web Link Graphic)算法.

输入：Web 链接图 G_w

输出：$Q(u_i,\langle u_i,u_j\rangle),Q(u_i)$

Begin

1. 初始化 G_w，对任意 $\langle u_i,u_j\rangle$，设置 $Q(u_i,\langle u_i,u_j\rangle)=0.0$.

2. 对 $Q(u_i,\langle u_i,u_j\rangle)$ 进行 k 趟更新操作. 设置计数器 $Counter=0$ 和更新标志 $bFlag=$false.

3. 取 G_w 中除叶节点外的任意一个在该趟尚未被计算的顶点(状态)u_i，使用式(17)计算 $Q(u_i,\langle u_i,u_j\rangle)$. 对于立即回报函数 $r(u_i,\langle u_i,u_j\rangle)$，如果 $u_j\in T$，立即回报取 rd(非零实常数)；否则立即回报取 0.0. 如果 u_i 的当前 Q 值与计算值不同，则更新 u_i 的 Q 值，并设置更新标值 $bFlag=$true. 否则，不作更新操作.

4. 如果当前 G_w 中除叶节点外的所有顶点在该趟都已经被计算一遍，转到步 5；否则回到步 3.

5. 判断更新标志 $bFlag$ 是否为 true，如果不是，则转到步 7；否则转到步 6.

6. 判断是否 $Counter=k$，如果是，转到步 7；否则 $Counter$ 加 1，设置更新标值 $bFlag=$false，转到步 3.

7. 对于 G_w 中的任意顶点 u_i，使用式(18)来计算 $Q(u_i)$.

8. 返回 $Q(u_i,\langle u_i,u_j\rangle),Q(u_i)$.

End.

对算法 3 来说，显然它满足收敛性的 3 个特定条件：(1) G_w 中每个状态转换过程是确定的，是一个确定性 MDP；(2) 立即回报值有界：rd 取 0.0 或一非零实常数；(3) G_w 具有有限状态，系统可对 G_w 的每个状态-动作对进行无限频繁地访问.

下面对算法 3 的算法复杂度进行分析. 设生成的 Web 链接图中：节点数为 n，叶节点数为 m，边数为 e，算法更新操作的趟数为 k. 每个节点的平均出度 l 定义为

$$l=\frac{1}{n}\sum_{i=1}^{n}O_i \tag{19}$$

其中，O_i 表示 G_w 中第 i 个节点的出度. 对算法 3 来说，由于很难准确估计一个 Web 链接图中每个节点的实际出度，所以这里使用了平均出度的概念.

(1) 时间复杂度.

从式(17)可知，对于任意一个状态-动作对 $\langle s,a\rangle$，计算一次 $Q(s,a)$ 共需要：计算一次回报值，求 l 个数中的最大值和一次加法计算. 又由算法 3 可知，一共需对 $(n-m)$ 个节点做 k 趟更新操作，设每个基本操作的时间复杂度为 $O(1)$，则算法的时间复杂度为 $O(k(n-m)l)$.

对于一个边稀疏的图来说，$l\ll n$，并且随着 n 增大，l 减少. 而 G_w 具有边稀疏的特性，通过对生成的多个 G_w 的统计，l 在 1～5 之间. 同时，相对与 n，$m\ll n$. 因此，算法 3 的时间复杂度实际上主要由节点数 n 和更新操作的趟数 k 决定. 最终算法的时间复杂度为 $O(kn)$. 考虑到在 k 趟更新操作的过程中，Q 值可能会提前收敛(算法 3 设置了更新标志)，所以实际的计算时间可能会更少.

(2) 空间复杂度.

算法 3 的空间复杂度主要体现在 G_w 的存储数据结构上. 在实际设计中，我们采用邻接表来表示 G_w. 该邻接表由 n 个顶点节点和 e 个边节点组成，因此，算法的空间复杂度为 $O(n+e)$. 由于邻接表中的顶点节点存储了有关该节点(链接)的特征信息(参见式(4))，属于一个复杂对象节点，且 G_w 具有边稀疏的特性，所以算法的空间复杂度主要由顶点节点的个数 n 决定，最终算法的空间复杂度为 $O(n)$. 为减少 G_w 节点数量的增加对系统内存空间的占用，我们采用了对象缓存(Cache)和序列化(serializable)技术来解决这个问题.

从以上对算法 3 的算法复杂度理论分析以及算法 3 的实际运行情况来看，算法 3 是可行的.

例 1. 为了说明以上的算法，这里给出一个计算实例. 在图 2 所示的 Web 链接图的例子中，给出了 10 个顶点和 11 条边，其中顶点 3、6、10 为叶节点. rd 设置为 100.00，$\gamma=0.9$. 在图 2 中，各条边所标注的数值是最终计算出的 $Q(u_i,\langle u_i,u_j\rangle)$ 值. 很显然，最后有：$Q(1)=90.0$，$Q(2)=100.0$，$Q(4)=90.0$，$Q(5)=100.0$，$Q(7)=90.0$，$Q(8)=90.0$，$Q(9)=100.0$.

为使式(3)中 $sim_{\text{predicted}}(u,t)$ 和 $sim_{\text{anchor}}(u,t)$ 具有可比性，应将 $sim_{\text{predicted}}(u,t)$ 的值映射到 $[0,1)$. 为此，只需将立即回报常数 rd 设置为 1.0 即可. 因此在本文中，rd 被设置为 1.0.

4 实验与分析

为了验证本文提出的 JLSE 算法的有效性，我们使用 Java 语言在 Eclipse 平台上实现了该算法的原型系统——JLSECrawler. 该原型系统的输入是特定领域的样本集合、一组种子 URL，输出是主题相关的结果页面集合. 实验环境为 CPU (Intel Celeron 2.66GHz)＋RAM (2GB)＋Window XP＋Eclipse3.4.

4.1 数据集

实验一共用到 4 个数据集：(1) 页面分类器训练样本集 *WebSet*1；(2) 结果页面分类器训练样本集 *WebSet*2；(3) 种子 URL 数据集 *URLSet*；(4) 目标页面 URL 数据集 *TargetSet*. *TargetSet* 主要用于评估聚焦爬行的目标召回率(参看 4.2 节).

前 3 个数据集为手工方式构建，而 *TargetSet* 为半自动方式创建. 为了比较算法在不同领域的爬行效果，我们分别在 3 个主题域(手机、数码相机、笔记本电脑)上进行了实验.

表 1 显示了数据集 *WebSet*1 的构成情况. 这里，需要对表 1 中 3 个主题域的反例数据来源做一说明. 在随机选择反例的过程中，为使反例更加具有代表性，应尽量使反例的选择较均匀地分布在除该主题域之外的其他主题域.

数据集 *WebSet*2 和 *URLSet* 的构建比较简单. *TargetSet* 分为 3 个子数据集，分别包含了与手机、数码相机和笔记本电脑有关的目标页面 URL. 表 2 给出了 *WebSet*2、*URLSet* 和 *TargetSet* 3 个数据集的情况.

表 1　页面分类器训练样本集 *WebSet*1

主题域	页面数量		数据来源	
	正例	反例	正例	反例
手机	1012	1003	http://mobile.sina.com.cn/ http://mobile.pconline.com.cn/ http://mobile.it168.com/ http://mobile.zol.com.cn/	除手机页面外，随机选择.
数码相机	1245	1229	http://tech.sina.com.cn/digital/ http://dc.pconline.com.cn/ http://dc.it168.com/ http://dcdv.zol.com.cn/	除数码相机页面外，随机选择.
笔记本电脑	1176	1142	http://tech.sina.com.cn/notebook/ http://notebook.pconline.com.cn/ http://notebook.it168.com/ http://nb.zol.com.cn/	除笔记本电脑页面外，随机选择.

表 2　*WebSet*2、*URLSet* 和 *TargetSet*

数据集	集合大小	数据来源
*WebSet*2	正例(细节页)：667 反例(链接页)：638	http://www.sina.com.cn/ http://www.163.com./ http://www.sohu.com/
URLSet	6 个 URL	http://www.pconline.com.cn/ http://www.chinadrtv.com/ http://www.amazon.cn/ http://www.pcpop.com/ http://www.gome.com.cn/ http://www.it168.com/
TargetSet	子集合①：10000个URL 子集合②：10000个URL 子集合③：10000个URL	使用Google搜索关键字："手机". 使用Google搜索关键字："数码相机". 使用Google搜索关键字："笔记本电脑".

4.2　实验设计

为了评价 JLSE 算法的实际运行效果，我们选取其它 4 种爬行策略进行对比实验. 这 4 种爬行策略分别是 BFS(Breadth-First Search)、ATA(Anchor Text Analysis)、CFC(Context Focused Crawler)[3] 和 OPIE[9]. 5 种算法中，除 BFS 外，其它 4 种均采用了聚焦策略. CFC 是一种经典的聚焦爬行算法，它利用环境图的知识指导爬行过程. 在本实验中，CFC 的参数设置为环境图的深度，设置为 4，并且每层最大文档数不超过 300；NB 分类器的个数为 5，队列数为 6. ATA 算法的设计除了在式(3)上与 JLSE 不同外，其他部分与 JLSE 完全相同. ATA 没有利用 Web 链接图的知识，只采用了直接证据分析，即 $sim(u,t)=sim_{\text{anchor}}(u,t)$. 设计 ATA 的目的是为了验证 JLSE 利用 Web 链接图知识的爬行效果.

由于是在真实的 Web 环境下进行实验，基于 Web 的规模，我们无法获取有关某个特定主题确切的相关文档集合，所以传统的信息检索指标如精度(precision)和召回率(recall)无法直接使用. 我们对实验结果的评价采用以下两个近似指标：

(1) 平均收益率(average harvest rate). 该指标用于模拟精度指标(precision). 其定义如下[9,16-17]：

$$harvest\_rate@N=\frac{1}{|V|}\sum_{i\in V}r_i \tag{20}$$

式中，$harvest\_rate@N$ 是指在 t 时刻，系统已爬取的页面数量为 N 时的收益率. V 是当前系统已经爬取的页面集合. r_i 代表页面 i 与主题的相似度，如果页面 i 与主题相似，则 r_i 取 1，否则取 0.

(2) 平均目标召回率(average target recall). 该指标用于模拟召回率指标，其定义如下[9,16-17]：

$$target\_recall@N=\frac{|T\bigcap C_t|}{|T|} \tag{21}$$

式中，T 是指目标页面 URL 的集合，在本文中该集合为数据集 *TargetSet*. C_t 代表系统已爬取的结果页面所对应的 URL 集合.

4.3　实验结果及分析

按照 4.2 节的实验设计，我们对 5 种爬行策略分别进行了系统实现，并在 3 个不同的主题域(手机、数码相机、笔记本电脑)上进行了实验. 其中，增量学习因子 cf 设置为 150. 实验结果如图 3～图 5 所示. 下面分别对实验结果进行分析.

(1) 平均收益率分析

图 3 显示了 5 种爬行策略在 3 个不同的主题域上取得的平均收益率情况. 图 3 横坐标表示系统已爬行的网页数量，纵坐标表示在系统已爬取的页面数量为 N 时的平均收益率. 从图 3 可以看到，5 种爬行策略随着 N 的增加均出现了下降的趋势，但不同策略下降幅度并不相同. 其中 JLSE 和 OPIE 的下降趋势较平缓，而 BFS 则较陡峭. 出现下降趋势的主要原因是聚焦主题具有很强的稀疏性，在爬行过程中，已发现的主题相似页面的数量和系统已爬取的页面数量的增量幅度不同，很显然前者是小于后者的，所以造成平均收益率总体上出现下降. 从图 3 中还可以看到，JLSE 的平均收益率最高，在 0.68～

0.55 之间. 由于使用了联合链接相似度评估的策略,使得很多具有延迟收益的链接能够被爬取,因此提高了平均收益率. 而 OPIE 平均收益率次之,在 0.65～0.51 之间. ATA 的平均收益率在 0.67～0.38 之间,CFC 在 0.57～0.31 之间. BFS 的平均收益率最低,仅在 0.26～0.03 之间. 由于 BFS 未采用聚焦策略,所以随着已爬取的页面数量的增加,其平均收益率下降得较快.

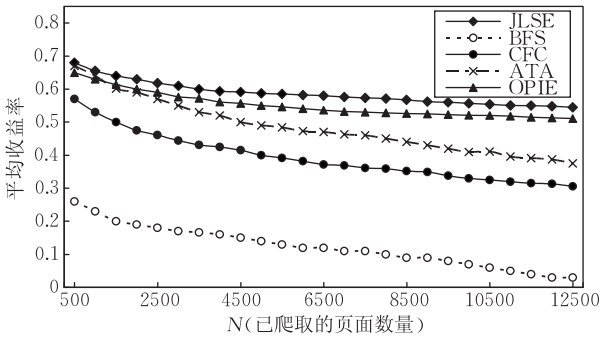

图 3 JLSE、BF、CG、ATA 、OPIE 5 种爬行策略在 3 个主题域上爬行时的平均收益率

(2)平均目标召回率分析

图 4 显示了平均目标召回率的情况. 图 4 横坐标表示系统已爬行的网页数量,纵坐标表示在系统已爬取的页面数量为 N 时的平均目标召回率. 从图 4 中,可以看到 5 种爬行策略随着 N 的增加出现上升趋势. 从式(21)可以看出,随着 N 的增加,$|T\cap C_t|$ 趋于上升,即系统爬取的主题相关的页面数量增加,因此平均目标召回率也随之增加,这是造成平均收益率总体上出现上升的原因. 而不同策略上升幅度不同的原因与平均收益率有关. 在 N 相同的情况下,平均收益率越高,$|T\cap C_t|$ 越大,则平均目标召回率越高. 从图 4 中可以看到,JLSE 的平均目标召回率最高,在 0.02～0.28 之间,而 OPIE 平均目标召回率次之,在 0.02～0.26 之间. ATA 平均目标召回率在 0.02～0.22 之间,CFC 在 0.02～0.18 之间. 而 BFS 平均目标召回率最低,仅在 0.01～0.1 之间. 从式(21)可知,增加系统的爬行时间,可以提高系统的平均目标召回率. 但当爬行时间超过一定阈值,$|T\cap C_t|$ 会趋于一个常数,所以平均目标召回率曲线会逐渐趋于与 X 轴平行. 这也与图 4 情况基本吻合.

另外,尽管 JLSE 的平均目标召回率最高,但也仅有 0.28. 造成平均目标召回率总体偏低的原因是:实验是在开放的 Web 环境下进行的,而与聚焦主题有关的网页分布具有很强的稀疏性. 从式(21)可知,平均目标召回率与数据集 *TargetSet* 的选取有较大关系. 因此,平均目标召回率的大小只具有相对意义. 而在 *TargetSet* 确定的情况下,平均目标召回率的大小与链接评估器的精度、实验提供的种子 URL 的数量、种子 URL 的代表性(覆盖性)、爬行时间等因素有关. 增加种子 URL 的数量、选择有代表性的种子 URL 以及增加爬行时间,均可以在一定程度上提高平均目标召回率. 但提高平均目标召回率的根本方法在于提高链接评估器的精度.

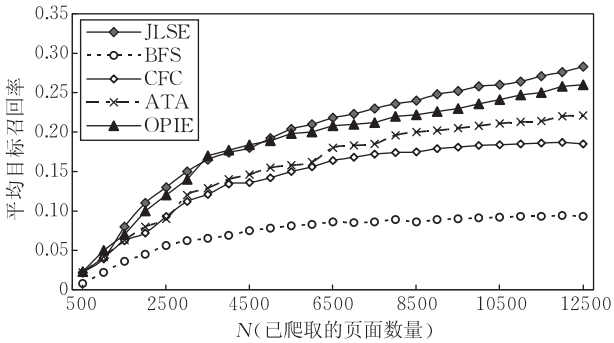

图 4 JLSE、BF、CG、ATA 、OPIE 5 种爬行策略在 3 个主题域上爬行时的平均目标召回率

(3)增量学习因子的影响

对 Web 链接图,系统采用了增量构建和学习的策略. 增量学习因子 cf 的取值对 Web 链接图生成的时间以及学习的效果会产生影响. 图 5 给出了不同的 cf 值对 JLSE 算法的平均收益率的影响. 图 5 横坐标表示增量学习因子的取值,纵坐标表示在系统已爬取的页面数量为 N 时的平均收益率. N 代表系统已爬取的页面数量. 从图 5 可以看到,对同一组 cf,随着 N 的增加,平均收益率均呈下降趋势,原因与前面的平均收益率分析相同. 另一方面,当 cf 分别取 50、100、150 时,随着 cf 增加,不同参数 N 的平均收益率曲线均呈上升趋势. 但当 cf 增加到 200 时,与 $cf=150$ 相比,所有的平均收益率曲线却都出现了下降趋势. 其原因可能是:一般说来,cf 取得越大,一次学习到的 Web 链接图的节点越多,那么链接学习器学习到的模型精度越高,这样会提高链接评估器的精度,其结果是提高了爬取到的结果页面的数量. 但如果 cf 取得太大,则会延长链接学习器进行下一次学习的时间间隔,这样又会降低链接评估器精度提高的速度,使得在这个时间段的爬行中,“漏掉”的目标页面的数量增加,从而降低了平均收益率. 所以在实际使用时,cf 的取值设置为 150 比较合适.

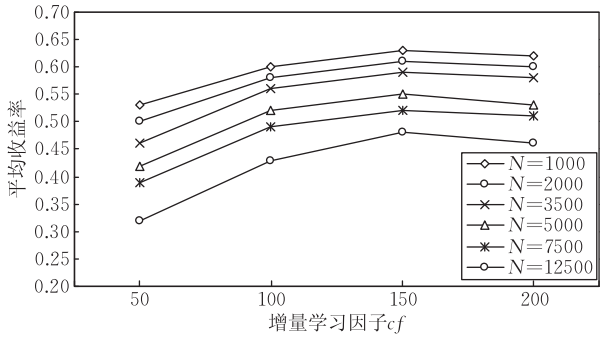

图 5　增量学习因子 cf 对 JLSE 在 3 个主题域上爬行时的平均收益率的影响

(4)学习 Web 链接图的作用

从图 3 和图 4 还能看到 JLSE 算法利用 Web 链接图的知识对爬行效果的影响. 从图 3 中,可以看到随着已爬行的网页数量的增加,ATA 的收益率下降得比较明显,总体上低于 JLSE 和 OPIE. 图 4 中也显示了类似的结果:ATA 的平均目标召回率低于 JLSE 和 OPIE. 因此可以说,利用 Web 链接图的知识对改善 JLSE 爬行效果的作用是明显的. 尽管学习 Web 链接图和计算联合链接相似度增加了系统的开销,但却较明显地提高了系统爬行的平均收益率和平均目标召回率,这种代价是值得的.

5　结　语

采用聚焦爬行策略从 Web 上获取感兴趣的资源是许多 Web 研究领域的热点问题. 本文提出一种基于联合链接相似度评估的爬行算法,该算法在评估链接的主题相似度时,同时使用了关于链接主题相似度的直接证据和间接证据. 通过在 3 个主题域上进行实验,我们发现该算法可以显著地提高爬行结果的精度和召回率.

然而本文提出的方法还存在一些不足,需要在未来的工作中加以解决. 首先,算法的过程比较复杂,涉及到很多在线的计算,这使得系统运行效率还有待提高. 下一步拟采用分布式计算的方法,将比较耗时的系统组件分布到多台机器上进行并行工作,以提高系统的计算效率. 其次,需要进一步提高页面分类器和结果页面分类器的工作效率和精度.

参　考　文　献

[1] Chakrabarti Soumen, van den Berg Martin, Dom Byron. Focused crawling: A new approach to topic-specific Web resource discovery. Computer Networks(CN), 1999, 31(11-16): 1623-1640

[2] Chakrabarti Soumen, Punera Kunal, Subramanyam Mallela. Accelerated focused crawling through online relevance feedback//Proceedings of the 11th International Conference on World Wide Web(WWW 2002). Honolulu, Hawaii, USA, 2002: 148-159

[3] Diligenti Michelangelo, Coetzee Frans, Lawrence Steve, Giles C Lee, Gori Marco. Focused crawling using context graphs//Proceedings of the 26th International Conference on Very Large Data Bases(VLDB 2000). Cairo, Egypt, 2000: 527-534

[4] Barbosa Luciano, Freire Juliana. An adaptive crawler for locating hidden web entry points//Proceedings of the 16th International Conference on World Wide Web(WWW 2007). Banff, Alberta, Canada, 2007: 441-450

[5] Rennie Jason, McCallum Andrew. Using reinforcement learning to spider the Web efficiently//Proceedings of the 16th International Conference on Machine Learning (ICML-99). Bled, Slovenia, 1999: 335-343

[6] Guilherme T de Assis, Alberto H F Laender, Marcos André Gonçalves, Altigran Soares da Silva. A genre-aware approach to focused crawling. World Wide Web(WWW), 2009, 12(3): 285-319

[7] Wang Hui, Zuo Wan-Li, Wang Hui-Yu, Ning Ai-Jun, Sun Zhi-Wei, Man Chun-Lei. Centroid-based focused crawler with incremental ability. Journal of Computer Research and Development, 2009, 46(2): 217-224(in Chinese)
(王辉, 左万利, 王晖昱, 宁爱军, 孙志伟, 满春雷. 基于质心向量的增量式主题爬行. 计算机研究与发展, 2009, 46(2): 217-224)

[8] Abiteboul S, Preda M, Cobena G. Adaptive on-line page importance computation//Proceedings of the 12th International Conference on World Wide Web (WWW 2003). Budapest, Hungary, 2003: 280-290

[9] Guan Ziyu, Wang Can, Chen Chun, Bu Jiajun, Wang Junfeng. Guide focused crawler efficiently and effectively using online topical importance estimation//Proceedings of the 31st Annual International ACM SIGIR Conference(SIGIR 2008). Singapore, 2008: 757-758

[10] Peng Tao, Meng Yu, Zuo Wan-Li, Wang Ying, Hu Liang. Tunneling techniques for focused Web crawling. Journal of Computer Research and Development, 2010, 47(4): 628-637(in Chinese)
(彭涛, 孟宇, 左万利, 王英, 胡亮. 主题爬行中的隧道穿越技术. 计算机研究与发展, 2010, 47(4): 628-637)

[11] Ahlers Dirk, Boll Susanne. Adaptive geospatially focused crawling//Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM 2009). Hong Kong, China, 2009: 445-454

[12] Yang Jiang-Ming , Cai Rui, Wang Chun-Song, Huang Hua, Zhang Lei, Ma Wei-Ying. Incorporating site-level knowledge

for incremental crawling of Web forums: A list-wise strategy// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2009). Paris, France, 2009: 1375-1384

[13] Alam Md Hijbul, Ha Jongwoo, Lee Sangkeun. Fractional PageRank crawler: Prioritizing URLs efficiently for crawling important pages early//Proceedings of the 14th International Conference on Database Systems for Advanced Applications (DASFAA 2009). Brisbane, Australia, 2009: 590-594

[14] Davison Brian D. Topical locality in the Web//Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2000). Athens, Greece, 2000: 272-279

[15] Mitchell Tom M. Machine Learning. Beijing: China Machine Press, 2003

[16] Pant G, Menczer F. Topical crawling for Business Intelligence//Proceedings of the 7th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2003). Trondheim, Norway, 2003: 233-244

[17] Pant Gautam, Srinivasan Padmini. Link contexts in classifier-guided topical crawlers. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2006, 18(1): 107-122

ZHANG Nai-Zhou, born in 1970, Ph. D. candidate. His main research interests include Web data management and information integration.

LI Shi-Jun, born in 1964, Ph. D., professor, Ph. D. supervisor. His main research interests include Web data management and information integration.

YU Wei, born in 1987, Ph. D. candidate. His main research interests focus on Web data management.

ZHANG Zhuo, born in 1978, Ph. D. candidate. His main research interests focus on Web data management.

Background

For many fields of Web research, how to fetch the interesting resources on Web has been an important research subject. At present, the chief method for obtaining the domain-specific resources on Web is to adopt the strategy of focused crawling.

The problem of focused crawling has been extensively studied. Now the main methods for focused crawling include the methods based on content analysis and the ones based on link analysis. However, due to the giant size of Web, the task for collecting domain-specific information has become increasingly challenging. For the most current techniques of focused crawling, there are still many problems in simultaneously meeting the high efficient crawl and the high quality of crawl results.

This work is inspired by a current research project, which aims to build a product search engine to capture and fuse the multi-aspect information related to the product or subject across the Web. To the research project, the first issue to be addressed is how to build an efficient general focused crawler which can be used for various product domains. For this purpose, in this paper, the authors present JLSE (Joint Link Similarity Evaluation) based algorithm, a new framework that solves the above problems. The authors' main contributions are chiefly embodied in three aspects: Firstly, propose an efficient machine learning based algorithm for link evaluator, which combines the direct evidence with indirect evidence on the topic similarity of a link; Secondly, the system has a high degree of automation. Except for some training samples, the system does not require any human intervention during the running. And thirdly, the system has a good reusability. As long as the configuration file of system is modified, the system can be applied to other product domains.

This work is supported by the National Natural Science Foundation of China under grant No. 60970018. So far, the research team has published more than 20 papers on Web search and mining.