Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

On-line topical importance estimation: an effective focused crawling algorithm combining link and content analysis*

Can WANG[†], Zi-yu GUAN, Chun CHEN, Jia-jun BU, Jun-feng WANG, Huai-zhong LIN

(*School of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China*)

[†]E-mail: wcan@zju.edu.cn

Received June 23, 2008; Revision accepted Nov. 4, 2008; Crosschecked June 10, 2009

Abstract: Focused crawling is an important technique for topical resource discovery on the Web. The key issue in focused crawling is to prioritize uncrawled uniform resource locators (URLs) in the frontier to focus the crawling on relevant pages. Traditional focused crawlers mainly rely on content analysis. Link-based techniques are not effectively exploited despite their usefulness. In this paper, we propose a new frontier prioritizing algorithm, namely the on-line topical importance estimation (OTIE) algorithm. OTIE combines link- and content-based analysis to evaluate the priority of an uncrawled URL in the frontier. We performed real crawling experiments over 30 topics selected from the Open Directory Project (ODP) and compared harvest rate and target recall of the four crawling algorithms: breadth-first, link-context-prediction, on-line page importance computation (OPIC) and our OTIE. Experimental results showed that OTIE significantly outperforms the other three algorithms on the average target recall while maintaining an acceptable harvest rate. Moreover, OTIE is much faster than the traditional focused crawling algorithm.

Key words: Focused crawlers, Topical crawlers, PageRank, Classifiers, On-line topical importance estimation (OTIE) algorithm
doi:10.1631/jzus.A0820481 **Document code:** A **CLC number:** TP391.3

INTRODUCTION

Focused crawlers, also called ‘topical crawlers’, are designed for topical resource discovery from the Web. A focused crawler ideally would download only the Web pages that are relevant to a given topic while avoid downloading all others. To better focus the crawling on a topic, a focused crawler will continuously update the priorities of unvisited uniform resource locators (URLs) in its crawl frontier during the crawling process. The main issue in focused crawling therefore is to effectively prioritize the currently uncrawled URLs in the frontier. There is a similar research issue in general crawlers in which the goal is to focus the crawler on the important portion of the whole Web since it is impossible for search engines to download the whole Web (Cho *et al.*, 1998).

Traditional focused crawlers exploit content (typically with a classifier) in the downloaded pages (i.e., link contexts) to predict relevance of the unvisited URLs and prioritize the frontier accordingly [we call this prioritizing algorithm ‘link-context-prediction’ (Pant and Srinivasan, 2006)]. However, content-based approaches usually demonstrate a poor performance in the Web. This is not only due to the noisy content in Web documents (i.e., documents with little text, containing images, scripts and other types of data that cannot be used by content-based approaches), but also because of different authorships of these documents and the lack of coherence in style or structure. Furthermore, content-based approaches are vulnerable to content spamming, which misleads search results ranking by using purposely tailored contents. Considering these issues, a more reliable heuristics is needed in focused crawling. Another source of information that can be exploited in Web information retrieval (IR) is the link structure of the Web.

* Project (No. 2007C23086) supported by the Science and Technology Plan of Zhejiang Province, China

Connectivity analysis of the Web is a well studied issue and several link-based metrics have been proposed (e.g., PageRank). These metrics measure the quality (or importance) of Web pages from different perspectives.

Our purpose is to assess the quality of uncrawled Web pages with respect to the concerned topic and to prioritize the frontier accordingly. This idea is supported by work in other research fields where link-based metrics in conjunction with content-based metrics were found to improve retrieval performance (Bharat and Henzinger, 1998; Silva *et al.*, 2000; Yang, 2001; Haveliwala, 2002; Calado *et al.*, 2003). However, these studies mainly focus on Web query results ranking or Web document classification. Web crawling, as a process of IR, has its particular constraints as summarized below:

(1) During the process of crawling, a crawler has only partial information of the Web. Unlike Web query results ranking and Web document classification, a crawler does not begin with a complete corpus, but a set of initial seed URLs. It is not until the end of the crawling process that the crawler has acquired knowledge of the complete graph.

(2) A crawler can follow only outward links of previously fetched pages to retrieve new pages. This means that a crawler can achieve only a local optimum by prioritizing the crawl frontier in each crawling iteration, while for the Web query results ranking a global optimal ranking of documents can be obtained in theory with all the information available.

(3) The on-line nature of crawling process poses a challenge to frontier prioritizing algorithms. Ideally, a focused crawler needs to reorder its frontier every time a page is fetched to select the most beneficial URL to crawl for the next iteration. If the time complexity of the prioritizing algorithm is high, the performance of the crawler will dramatically decrease. Scalability of the frontier prioritizing algorithm is therefore an important issue since a focused crawler will typically need to repeatedly rank a huge number of URLs during its crawling process.

Considering the above constraints, we propose a new frontier prioritizing algorithm named ‘on-line topical importance estimation’ (OTIE) based on our previous work in (Guan *et al.*, 2008). As a scalable algorithm, OTIE takes both link and content evidence into account. In particular, it propagates a topical quality score between Web pages via hyperlinks on the crawled pages and uses the accumulated topical quality scores to prioritize the crawl frontier. We perform real crawling experiments to show OTIE’s advantages over traditional prioritizing algorithms.

PREVIOUS WORKS

The concept of focused crawling was first introduced by Menczer (1997) and Chakrabarti *et al.* (1999). Chakrabarti *et al.*(1999) proposed a hierarchical topic taxonomy with training examples and trained a naive Bayes classifier on this taxonomy. During the crawling process, they used this classifier to compute the relevance score (its value in [0, 1]) of the currently fetched page to the topics selected by the user, and then assigned this score to the unfetched URLs extracted from this page as their ranking scores in the frontier. This is based on the assumption that relevant pages are likely to introduce other relevant pages (Davison, 2002).

Most focused crawling algorithms are variations of the best-first-search (BFS) algorithm in which the crawl frontier is maintained as a priority queue. Each URL in the frontier has an associated ranking score. Different best-first crawlers may be produced by varying the heuristics used to score the uncrawled URLs.

The naive best-first crawler (Menczer *et al.*, 2001) measures and applies the cosine similarity of a crawled page to a topic profile or a query to estimate the benefit of following the hyperlinks found on that page. Pant and Srinivasan (2005) investigated the use of machine learning algorithms in the naive best-first crawler to estimate the similarity of a crawled page to the topic. By comparing three types of classifiers—naive Bayes, neural network, and support vector machine, they found that naive Bayes is a weak choice for guiding a focused crawler. In their following work (Pant and Srinivasan, 2006), they examined the effects of various definitions of link contexts on the crawling performance [the naive best-first crawler and the focused crawler in (Chakrabarti *et al.*, 1999) can be viewed as treating the entire page content as the context of the hyperlinks in the page].

Some complicated heuristics have been used to guide topical crawlers. The notion of context graph was introduced by Diligenti *et al.*(2000), in which a link-based ontology is required in the training phase. Another similar method is to use reinforcement learning (Rennie and McCallum, 1999), where the crawler is trained using paths, leading to relevant goal pages. An intelligent crawler that can learn its way on-line into the topic defined by a user using arbitrary predicates was discussed in (Aggarwal *et al.*, 2001). User browsing patterns have also been exploited to guide focused crawlers (Aggarwal, 2002).

Cho *et al.*(1998) suggested using partial PageRank (i.e., PageRank computed on the Web graph that is seen so far by the crawler) to prioritize frontier. However, this strategy proved to perform poorly in focused crawling (Menczer *et al.*, 2001; Chau and Chen, 2003). This is because PageRank is a topic-free metric and some universally important pages will always gain high scores and mislead the crawler. Nevertheless, researchers were attempting to incorporate link-based analysis into focused crawling. In the well-known foundational work, Chakrabarti *et al.*(1999) incorporated periodical topic distillation into the crawling process which implements a modified version of Kleinberg's hyperlink-induced topic search (HITS) algorithm (Kleinberg, 1998) to identify good hub pages. These hubs were then revisited and the priorities of the unvisited URLs cited by them were raised. Chau and Chen (2003) modeled the Web graph as a neural network and incorporated a spreading-activation algorithm to compute the weight value for each node. The link weight of a link pointing from page *i* to page *j* was set to the relevance of page *j* to the target topic inferred from clues in page *i*. Almpanidis *et al.*(2005) modified the latent semantic indexing (LSI) technique by incorporating link information into the term by a document matrix. They employed cosine similarity between the projected topic vector and projected vectors of unvisited pages to prioritize these pages. Jamali *et al.*(2006) maintained a set called 'seed pages' in crawling. Fetched pages were ranked according to their connectivity with the seed pages and also their similarities to the topic. At the end of each crawling loop the fetched page with the highest score was added as a seed page and hyperlinks within it were scheduled in the next loop.

However, the hybrid methods discussed above have one common problem. That is, they all need to maintain some form of data structure of the Web graph crawled so far and to apply the heuristic algorithm periodically to the data structure. As the crawling proceeds, the Web graph visited will grow continuously and so will the time needed to execute the algorithm and the memory needed to store the Web graph, which hinders the scalability of the crawler.

In OTIE, the prioritizing algorithm is naturally incorporated into the crawling process by computing the hybrid ranking scores in an on-line fashion and requires no storage of linkage information. Compared with the previous hybrid methods, less time and storage are needed in OTIE, making the crawling process highly efficient and scalable.

LINK- AND CONTENT-BASED ANALYSIS

The OTIE algorithm features a combination of link- and content-based analysis to effectively guide focused crawling. However, due to the three constraints previously mentioned, link-based techniques are difficult to be incorporated into focused crawlers although their value in Web IR is well recognized (Bharat and Henzinger, 1998; Silva *et al.*, 2000; Yang, 2001; Haveliwala, 2002; Calado *et al.*, 2003). The OPIC algorithm (Abiteboul *et al.*, 2003) that performs on-line computation of PageRank gave us the inspiration to incorporate content- and link-based analysis into focused crawling. To better explain the design of the OTIE crawler, we briefly review PageRank and OPIC, together with a discussion on the techniques used for content analysis.

Page importance metric: PageRank

The PageRank algorithm discussed in (Page *et al.*, 1998) uses the link structure of the Web to compute a rank vector, which gives a priori importance estimates to all of the pages on the Web. The rank vector is computed once, off-line, and is independent of the content of Web pages. The basic idea behind PageRank is that if page *i* has a link to page *j*, then the author of page *i* implicitly confers some prestige to page *j*. The importance of a page is defined recursively as the weighted sum of the importance of the pages pointing to it. The process of computation of PageRank scores can be expressed in terms of matrix eigenvector calculation. Let $\boldsymbol{r}$ denote the PageRank vector, O_p the set of pages corresponding to out-links of page p, $\boldsymbol{A}$ the adjacency matrix of the directed graph G of the Web, and $\boldsymbol{M}$ a variation of $\boldsymbol{A}$ in which the ith row vector is equal to the ith row vector in $\boldsymbol{A}$ divided by $|O_i|$. $\boldsymbol{r}^*$ is the solution to

$$\boldsymbol{r}=\frac{\boldsymbol{M}^{\mathrm{T}}\times\boldsymbol{r}}{\left\|\boldsymbol{M}^{\mathrm{T}}\times\boldsymbol{r}\right\|_1}. \tag{1}$$

$\boldsymbol{M}^{\mathrm{T}}$ can be viewed as the stochastic transition matrix over G and $\boldsymbol{r}^*$ as the stationary probability distribution over pages induced by a random walk on the Web. However, the convergence of PageRank computation is guaranteed only if G is strongly connected and is aperiodic. The latter condition is practically satisfied in the Web graph, while the former can be satisfied by adding a ‘small’ edge of weight α/N between every pair of nodes in the graph, where N is the number of Web pages and α is a damping factor indicating the probability that a surfer will continue on clicking on a hyperlink in the page (Haveliwala, 2002). Therefore, the real matrix used in the computation of PageRank is

$$\boldsymbol{M}'=(1-\alpha)\boldsymbol{M}+\alpha[1/N]_{N\times N}. \tag{2}$$

PageRank is a global, static measure of quality of a Web page and claims applications in global-scale search engines.

OPIC

Abiteboul *et al.*(2003) proposed an algorithm called OPIC (on-line page importance computation), which computes PageRank in an on-line fashion and requires fewer resources. The algorithm maintains only two values, cash and history, for each page. Initially, the total amount of cash is evenly distributed to each page. While the algorithm runs, the cash variable of a page records the sum of cash obtained by the page since the last time it was crawled, and the history variable of a page records the sum of cash obtained by the page since the start of the algorithm until the last time it was crawled. The algorithm continuously selects pages to process. When a page is selected, its cash is distributed equally among the pages it points to, added to its history, and then reset to 0. To estimate PageRank scores, Abiteboul *et al.* (2003) defined a vector variable $\boldsymbol{x}_t$ in the scene of the end of the tth iteration of the algorithm as follows:

$$\boldsymbol{x}_t=\frac{\boldsymbol{h}_t}{\left\|\boldsymbol{h}_t\right\|_1}, \tag{3}$$

where $\boldsymbol{h}_t$ is the vector of history of all pages at the end of the tth iteration. It can be proved that $\boldsymbol{x}_t$ converges to the real PageRank vector as t goes to infinity. To guarantee convergence, a virtual page is introduced to point to and get pointed to by all other pages.

The algorithm does not impose any requirement on the order in which the pages are visited, as long as each page is visited infinitely often. Abiteboul *et al.* (2003) evaluated three page selection strategies—random, greedy, and cycle—with regard to their impact on the converging speed of the algorithm. The random strategy chooses the next page to crawl randomly, while the greedy strategy chooses the next page with the highest cash, and the cycle strategy selects pages in a fixed order and cycles around. By conducting experiments with a synthetic graph they found the greedy strategy to be the best among the three strategies with respect to the converging speed of the algorithm.

Abiteboul *et al.*(2003) also considered the situations where the graph is changing and proposed an adaptive version of OPIC that stays close to the changing PageRank scores. Interested readers are referred to (Abiteboul *et al.*, 2003) for details.

Techniques for content analysis

We use the well-known vector space model (VSM) to represent a Web page p as an n-dimensional vector $\boldsymbol{x}_p=[w_{1p}, w_{2p}, \ldots, w_{np}]$, where n is the number of terms in the vocabulary, and w_{ip} is the weight of term i in page p. Term frequency - inverse document frequency (TF-IDF) is a common method for weighting terms in a document:

$$w_{ip}=f_{ip}\times\ln(|E|/f_{\mathrm{d}i}), \tag{4}$$

where f_{ip} is the frequency of term i in page p, E is the set of documents, and $f_{\mathrm{d}i}$ (document frequency, or DF) is the number of the pages in E that contain the term i.

For our experiments, topics are obtained from the Open Directory Project (ODP) (http://www.dmoz.org) and classifiers are used for content analysis. The training data are obtained from Web pages corresponding to the URLs that belong to the selected topics in ODP. For each topic, positive examples are Web pages of this topic, and negative examples are randomly selected Web pages from other selected topics. The number of negative examples is always twice the number of positive examples. After HTML tags are removed, examples are represented in TF-IDF vector space using standard text processing techniques (tokenization, stop words removal, stemming, etc.). Firstly, the training set is generated by randomly selecting a predefined number of pages from the positive examples and twice the number from negative examples. Other examples are kept for testing the trained classifiers. Then we derive the vocabulary V for the topic from the training set. We apply the χ^2-test feature selection method to V. With the vocabulary, we then can represent all the examples (and arbitrary text content) as TF-IDF vectors using Eq.(4), where the document collection E is the training set.

The converted examples are then used to train and test classifiers, of which three types are used: support vector machine (SVM), naive Bayes (NB) and neural network (NN). We use the SVM classifier to estimate the conditional probability $Pr(class=+1|\boldsymbol{x}_i)$ for page i (also text snippets around hyperlinks) as its similarity estimation, and use all the three classifiers to classify Web pages. Parameters with which these classifiers (and some other parameters mentioned above) are trained are later illustrated in the EXPERIMENTS AND RESULTS section. Due to limited space we refer readers to (Jain *et al.*, 1996; Elkan, 1997; Burges, 1998) for details on the classifiers.

OTIE

In this section, we first briefly outline the architecture of our focused crawler and then explain the design of the OTIE algorithm in detail.

Crawler architecture

We implement our focused crawler based on a general-purpose crawler used by an open source search engine named Nutch (http://lucene.apache.org/nutch/). The system is designed so that any specific prioritizing algorithm can be easily plugged into the system through a standard interface. Fig.1 illustrates the architecture of the focused crawler. The topic repository contains topic-specific resources, including seed URLs to initiate a crawl and examples for training classifiers. The crawl DB stores all the URLs the crawler has seen so far, including URLs in the crawl frontier and URLs of the downloaded pages. The page repository contains all the downloaded pages. In each iteration of the crawling loop, a list of URLs to be fetched (URL fetch-list) is generated from the crawl frontier according to their ranking scores. The URL fetch-list is a fixed-length queue-like data structure shared by all worker threads, which are responsible for fetching, parsing and storing Web pages. Worker threads are also responsible for scoring unvisited URLs found on the downloaded pages. At the end of each iteration, all out-links extracted from the downloaded pages are merged into the crawl DB (in OTIE, scores of duplicate URLs are accumulated while in link-context-prediction a lower score is overwritten by a higher one). Our focused crawler implements a best-N-first crawler, which generally outperforms a strict best-first crawler (Pant *et al.*, 2002). Here N is the length of the URL fetch-list. The crawling process continues until a sufficient number of pages are fetched or there is no unvisited URL. In practice, a focused crawler usually stops when a threshold of crawled pages is reached. The URLs with low ranking scores will never be crawled since they always stay at the tail of the crawl frontier. A poor ranking algorithm may keep good URLs at the tail of the crawl frontier, leaving important pages uncrawled.

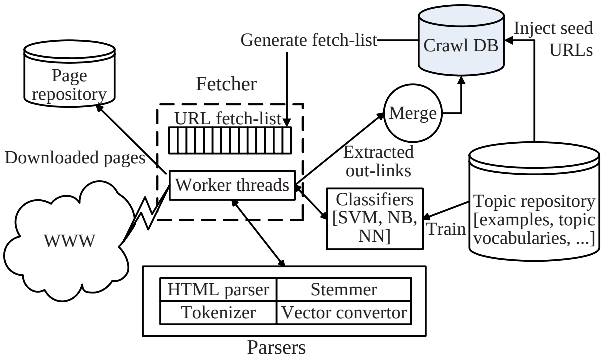

Fig.1 Architecture of our focused crawler

Design of the OTIE algorithm

The OTIE frontier prioritizing algorithm used in our focused crawler is inspired from the OPIC algorithm. Like in OPIC, we transfer 'cash' (i.e., importance) among pages in OTIE. Since pure link-based methods prove to perform poorly in focused crawling (Menczer *et al.*, 2001; Chau and Chen, 2003), we bias the cash distribution in OTIE to favor on-topic pages and to suppress off-topic pages. Let O_p denote the set of pages that page p points to. When a page i is crawled, we distribute its cash in accordance with the similarity scores of the pages in O_i with respect to the concerned topic; that is to say, the cash that page j in O_i gains from page i is proportional to its similarity to the topic:

$$\forall j \in O_i, \; cash\_gain(j,i) = \frac{sim(j,t)}{\sum_{u\in O_i} sim(u,t)} \times cash[i], \quad (5)$$

where *cash_gain*(j, i) represents the amount of cash that page j gains from page i, *sim*(j, t) represents the similarity of page j to topic t, and *cash*[i] denotes the amount of cash possessed by page i when it is being crawled.

Intuitively, biasing is meaningful. Biasing the distribution of cash means that the universally popular but irrelevant pages may receive relatively little cash, while the pages that are both important and relevant can be greatly rewarded. This would prevent the focused crawler from drifting away from the topic, and encourage it to download first the important pages as concerned with the topic.

Nevertheless, when the context is Web crawling, things become a little complicated. This is because we cannot foresee the content of uncrawled pages and consequently are not able to evaluate their similarities to the topic. This is also the main problem in traditional context-based focused crawlers. So, for an uncrawled page j, we substitute predicted similarities deduced from the link contexts in parent pages for the real similarity *sim*(j, t), yielding the following formula:

$$\forall j \in O_i, \; cash\_gain(j,i) = \frac{sim'(j,t)}{\sum_{u\in O_i} sim'(u,t)} \times cash[i], \quad (6)$$

where the similarity function *sim'*(j, t) is defined as

$$sim'(j,t) = \begin{cases} sim(j,t), & j \in S_{\text{fetched}}, \\ predicted\_sim(j,t), & j \in S_{\text{unfetched}}. \end{cases} \quad (7)$$

Here *predicted_sim*(j, t) is the similarity of page j's link context in the currently crawled page to topic t. S_{fetched} denotes the set of pages that the crawler has downloaded, while $S_{\text{unfetched}}$ represents the set of uncrawled pages in the frontier.

Interestingly, OTIE is more vulnerable to noisy link context than traditional focused crawlers that solely rely on link context to guide focused crawling. This is because child pages inherit cash from their parent pages and if an irrelevant page gains more cash, the whole sub-tree rooted at that page will be influenced. Preliminary experiments confirm our analysis: the crawler is still likely to be trapped in irrelevant areas of the Web graph.

To remedy this problem, we introduce a revision phase into the algorithm: when a page is crawled, the cash value is revised according to the following formula before distributing the cash:

$$\Delta cash = cash \times \max(-1, a(2r-1)^d), \quad (8)$$

where r is the relevance score of the currently crawled page with respect to the topic; a, d are parameters satisfying constraints: $a>0$, $d\in\{d|d>0$ and x^d is an odd function$\}$. Because r is in range [0, 1], the second argument of function max() is in range [$-a$, a]. Using this formula, pages with low similarities to the topic (i.e., $r<0.5$) are punished by reducing their cash, while pages with high similarities (i.e., $r>0.5$) are rewarded by increasing their cash. Reduction is bounded by cash to make sure that cash is always non-negative. Parameter d controls the flatness of the curve of function $a(2r-1)^d$ when r is near 0.5. Increasing d reduces the impact of revision on all values of r except in the extreme cases (when r=0 or r=1) and setting d to values above 1 makes revision relatively focusing on the pages that are particularly relevant or irrelevant. This is what we want because we do not intend to abruptly change the cash values of ambiguous pages. Parameter a controls the absolute degree to which cash is altered. By introducing the revision phase, however, the total amount of cash becomes a variable. Because our aim is not to estimate topical PageRank scores of pages, but to effectively guide focused crawlers, this is not a big problem.

Algorithm 1 OTIE prioritizing algorithm

```
OTIE()
 1 Inject seed URLs into frontier, with cash distributed evenly;
 2 while (frontier is not empty and fetched<MAX_PAGES)
 3   fetch_list=URLs in frontier with the highest priorities;
 4   if (the periodical condition is satisfied)
 5     Add the fetched URL with the highest cash into fetch_list;
 6   endif
 7   Download URLs in fetch_list;
 8   foreach ($p_i$ in successfully downloaded pages)
 9     Revise $p_i$'s cash using Eq.(8);
10     foreach ($p_j$ linked to by $p_i$)
11       $cash[p_j]=cash[p_j]+cash\_gain(p_j, p_i)$;
12     endfor
13     $cash[p_i]=0$.
14   endfor
15 endwhile
```

As shown in Algorithm 1, the OTIE algorithm goes as follows: initially, the total amount of cash is equally distributed among the set of seed pages (line 1). Afterward in crawling, when a page is crawled, its cash is revised according to Eq.(8) (line 9), and then according to Eq.(6), distributed among pages corresponding to the hyperlinks found on that page (lines 10 and 11). The priority of an uncrawled page is the amount of cash it possesses. Periodically, the previously fetched page with the highest cash is recrawled to distribute its cash (lines 4 and 5). The crawling process continues until a sufficient number of pages are fetched or there is no unvisited URL.

EXPERIMENTS AND RESULTS

Topic selection

We followed similar rules to that proposed in (Srinivasan *et al.*, 2005) to derive topics from ODP. Specifically, we selected topics at the same *TOPIC_ LEVEL*, which is defined as the distance from the root of the concept hierarchy to the node describing the topic. Srinivasan *et al.*(2005) suggested building topics from sub-trees of a given maximum depth (*MAX_DEPTH*) whose roots are the nodes corresponding to those topics, respectively. In our experiments we chose topics at a high *TOPIC_ LEVEL* (i.e., topics far from the root of the concept hierarchy) and did not impose the *MAX_DEPTH* constraint on the sub-tree for a topic. The reason is that, topics in the concept sub-tree rooted at a specific topic node tend to be conceptually clustered and thus can be treated as a whole to represent the root topic. Furthermore, the more specific the topics, the fewer the external links (from now on referred to as resource links of the topic) in the concept sub-trees that get rooted at these topics; thus, if restricting the depth of sub-trees we would obtain an insufficient number of external links.

The process used to extract topics from ODP is presented as follows:

(1) Download the resource description framework (RDF) format files containing concept hierarchy information (structure file) and information about each topic (content file) from the ODP Web sites.

(2) From the structure file randomly select 30 topics at *TOPIC_LEVEL*=3 (regard 'TOP' as *TOPIC_ LEVEL*=0). Topics under the general topics of Regional, Kids and Teens, World and Adults are excluded. This avoids many semantically similar topics, as well as pages with non-English content. We also filtered out topics with fewer resource links than those required by the training set. Two other constraints were imposed on the random selection process: no two selected topics have the same parent topic and no four selected topics have the same grandparent. The former constraint further avoids semantically similar topics, while the latter makes the selected topics diverse.

(3) For each selected topic extract its resource links from the content file to form the relevant set for that topic. The relevant set was divided into two random disjoint subsets. The first set, called 'seed set', was used to initiate the crawl; the second one, called 'target set', was used to evaluate crawlers. The seed set contained 10 URLs and relevant sets were also used to generate training sets and test sets for topics.

Table 1 shows two selected sample topics and their corresponding seed sets and target sets.

Algorithms to be evaluated

Real crawling experiments were conducted over the 30 topics selected from ODP. We evaluated four different prioritizing algorithms: breadth-first, link-context-prediction, OPIC (greedy strategy), and OTIE. Breadth-first was used as a baseline, since it makes no use of any knowledge about the topic. Its performance was expected to provide a lower bound for any intelligent focused crawling algorithms. The other three algorithms can be viewed as variations of the best-*N*-first-search (BNFS) algorithm. Here we set

Table 1 Sample topics and their corresponding seed sets and target sets

ODP topic	Seed set	Target set
Top/Health/Home_ Health/Home_Care	http://familychoicehc.com/	http://www.partoflife.com/
	http://www.advantage-nursing.com/	http://www.nyshcp.org/
	http://qualitycares.com/	http://www.kshomecare.org/
	http://www.willcare.com/	http://www.hcaoa.org/
	http://www.bloodtest.ca/	http://www.rcss.org.uk/
	http://members.aol.com/toyhorse50/	http://www.forestcomputer.com/kk/
	http://www.optionsforcare.com/	http://www.zoehomecare.com/
	http://www.companionconnectionseniorcare.com/	http://www.healingpalmskauai.com
	http://www.caringroad.org/index.cgi	http://www.caregiver.ca/
	http://www.northwesthealthcare.com/	…
Top/Shopping/ Tobacco/Cigars	http://www.tobaccostation.com.au/	http://www.abccigar.com/
	http://www.cigarfactoryneworleans.com/	http://www.barlowscigars.com/
	http://www.cigarworld.org/	http://www.cigar-connoisseur.co.uk/
	http://www.littlehavanatrading.com/	http://www.abnersworld.com/
	http://www.cigarsofcuba.co.uk/	http://www.blackcatcigars.com/
	http://www.nickscigarworld.net/	http://www.cigarstudio.com/
	http://www.bestvaluecigars.com/	http://www.avilacigars.com/
	http://www.heirloomhumidors.com/	http://www.epuff.com/
	http://www.aficionadocigar.com/	http://www.neptunecigar.com/
	http://www.cigars.com/	…

the parameter N to 50. For link-context-prediction, we adopted a hybrid definition of link context, which showed good performance in the context of focused crawling (Pant and Srinivasan, 2006):

$$score=\beta\times page\_score+(1-\beta)\times context\_score, \quad (9)$$

where *page_score* is the relevance score of the entire page content, *context_score* is the relevance score of a text window around a hyperlink on that page, and β is the relative weight assigned to *page_score*. We set β=0.25 and text window size T=20 words (including the anchor text). An OPIC-driven crawler was evaluated to confirm that ‘biasing’ leads to a significant performance improvement. Finally, when predicting the relevance score of an uncrawled page in OTIE, Eq.(9) was used again and the parameters remain the same as in link-context-prediction.

Metrics and settings

As we could not get the whole Web labeled regarding a topic, we used two surrogate metrics, i.e., harvest rate (Pant and Srinivasan, 2005; 2006) and target recall (Pant and Srinivasan, 2005; 2006; Srinivasan *et al.*, 2005), to approximate the two standard IR evaluation metrics, precision and recall, respectively. Harvest rate is defined as the fraction of crawled pages that are judged to be relevant. As human efforts are prohibitively large for judging a large number of Web pages, we depend on the three classifiers to make relevance judgment. A crawled page is considered to be relevant if at least two of the classifiers judge it to be relevant. Target recall is essentially the recall of the target set. Because URLs in target sets are manually collected and distilled by the ODP editors, to some degree they can be regarded as high quality resources for corresponding topics and then target recall as a metric indicating the ability of focused crawling algorithms in harvesting high quality topical Web pages.

We put an emphasis on the target recall metric in this study for the following two reasons: first, harvesting Web pages that are both of high quality and topically relevant is more important than just harvesting literally relevant ones for many topical resource discovery applications (Tang *et al.*, 2005); second, the target recall is to some degree a more reliable indicator than the harvest rate in that the classifiers used in measuring the harvest rate are prone to errors due to the noisy Web contents.

We kept the number of positive examples in the training set at 75, and thus the total number of training examples for a topic was 225. When generating the vocabulary, we selected the best 2000 words from the candidates. For the SVM classifier, we chose the first degree polynomial kernel as it showed good performance in guiding context-based crawlers (Pant and Srinivasan, 2005). The NB classifier was trained using kernel density. For the network structure of the NN classifier we chose the classical three-layer feed-forward network with four hidden nodes. The learning rate was set to 0.3, and a maximum of 500 training epochs were performed. We also set aside 30% of the training data for validation.

Eq.(8) has two parameters to be determined. We carried out preliminary experiments crawling 10 000 pages in each run for three topics to investigate the impact of different parameter values on the performance of OTIE (Fig.2). In particular, we first kept d=3.0 and tested a=0.50, 0.85, 1.00 and 1.50. Then a was fixed at 0.85 and d took values of 1.0 and 5.0. All the three topics showed similar evidences of the impact of varying the two parameters. Considering N=10 000, increasing a tends to increase the harvest rate, but to decrease the target recall (however, when a is small enough, decreasing a seems to decrease both of the two metrics, e.g., a=0.85 vs a=0.50), while varying the value of d seems not to influence the two performance metrics. In this study we set a=0.85 and d=3.0.

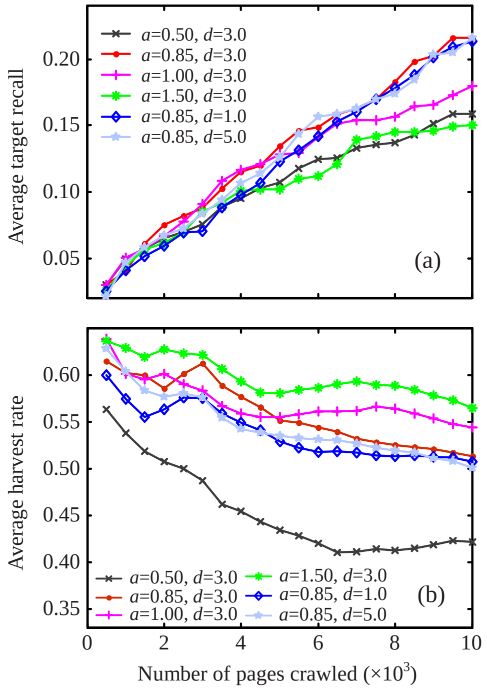

Fig.2 Target recall for biased-OPIC (a) and harvest rate for OTIE (b) with different revision parameters averaged over three topics during the crawling process

Results

Most of the previous research in topical crawling has been limited to 1000~25 000 page crawls and/or a small number of topics (Pant and Srinivasan, 2006). Consequently, we decided to use 30 topics to evaluate the four algorithms, with 20 000 pages crawled for each distinct (algorithm, topic) pair. Thus, a total number of 2.4 million pages were crawled. All crawls were performed in the same environment (i.e., computer hardware and network bandwidth). The two performance metrics were computed at different points during the crawl. We then calculated the mean harvest rate and mean target recall at these points over all the topics involved.

The experiment results are given in Fig.3, where the horizontal axis is the number of pages crawled, and the vertical axis represents the average target recall or average harvest rate. We also computed the ±1 standard errors for all the data points, as represented by the error bars in the plots. We performed one-tailed paired t-tests using significance level α=0.05 at N=20 000 to evaluate the significance of performance differences between prioritizing algorithms. We found that OTIE significantly

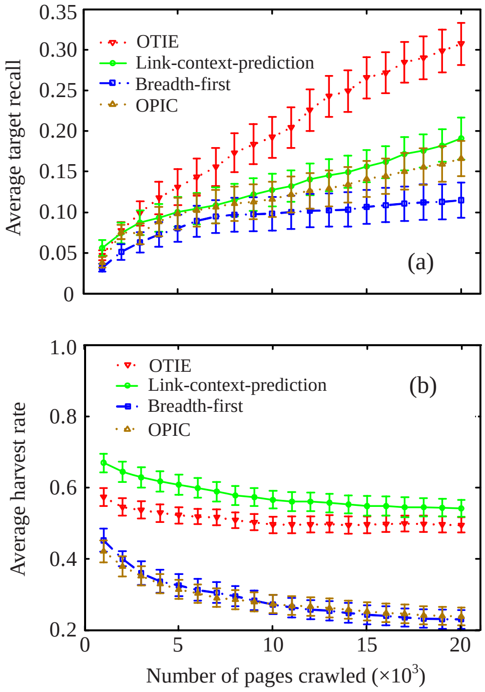

Fig.3 Target recall (a) and harvest rate (b) of the four prioritizing algorithms averaged over 30 topics during the crawling process

The error bars represent the ±1 standard errors outperformed the other algorithms on the average target recall; as for the average harvest rate, both link-context-prediction and OTIE showed significantly better performance than the other algorithms. Breadth-first and OPIC demonstrated almost equally poor performance on the average harvest rate.

Note that in the early stage of the crawling process, link-context-prediction's recall was the highest, but at the point near N=2000, the curve of OTIE went beyond the curve of link-context-prediction, and in the end, link-context-prediction was only marginally better than OPIC. As mentioned earlier, target recall measures the ability of focused crawling algorithms in harvesting high quality topical Web pages. This indicates that OTIE can retrieve high quality topical pages with a high efficiency. The initial poor performance of OTIE and OPIC can be explained by the lack of link evidence. Although harvest rate is not considered as important an indicator as target recall, we observed that the harvest rate curve of OTIE approached that of link-context-prediction as more link evidence was seen. In the end the performance difference was no longer significant.

When conducting experiments, we found another advantage of OTIE and OPIC over link-context-prediction. That is, OTIE and OPIC required much less time in crawling 20000 Web pages than link-context-prediction did. We did not include breadth-first into this comparison because it involves no frontier-reordering during its crawling process. Actually we observed that with the increasing size of the frontier, the frontier reordering time would gradually dominate the crawling loop. We selected 10 topics from the 30 topics to crawl 100000 pages for each (algorithm, topic) pair to further investigate the running time of the three algorithms for long crawls (for statistical results, refer to Table 2). The fact is that although our focused crawler implements multithreading in the fetcher component (Fig.1), the worker threads can still be used with a very low efficiency because the crawler is bound to the restriction about being polite with Web sites. In the extreme case, if in an iteration of the crawling loop the URLs to be fetched in the URL fetch-list are all from the same Web site, only one worker thread is used in this iteration, with all the other worker threads left unused. It seems that link-context-prediction is more likely to be trapped in a literally more relevant site, while by exploiting link evidences OTIE and OPIC tend to select more URLs belonging to different sites into the URL fetch-list, and thus make the use of worker threads more efficient.

Table 2 Comparison of the prioritizing algorithms in terms of average running time over the selected 10 topics when 100000 pages are downloaded*

Algorithm	Average running time±σ_M (min)
OTIE	1437.93±102.31
OPIC	1476.57±84.43
Link-context-prediction	4115.73±156.13

* The length of URL fetch-list is 50, and 30 worker threads are used. σ_M: standard error

CONCLUSION

In summary, the OTIE algorithm proposed in this paper effectively combines link- and content-based analysis to evaluate the benefit of following an uncrawled URL. Compared with previously proposed prioritizing algorithms for focused crawling that make use of both link evidence and content evidence, our algorithm is much faster and requires less storage in that, instead of introducing an exclusive computing phase and data structures for the algorithm, we incorporate it seamlessly in the crawling process. We conducted experiments on the 30 topics selected from ODP and used a specialized version of the evaluation framework for focused crawlers proposed in (Srinivasan *et al.*, 2005) to evaluate four different prioritizing algorithms. Results showed that OTIE significantly outperforms the other three algorithms on the average target recall, indicating its excellence at harvesting high quality topical Web pages. Moreover, OTIE is faster than link-context-prediction in terms of the time used to download the same number of pages.

More experiments on topics other than those used in this work are obviously needed. Part of our future work will be focused on this issue. In this study we adopt PageRank as the page importance metric. There are other page importance metrics in the literature, such as hub and authority in Kleinberg's HITS algorithm. The hub metric was found to boost the performance of focused crawlers (Pant and Srinivasan, 2006). Another direction for our future work is to investigate the performance of OTIE when page importance metrics other than PageRank are used.

References

Abiteboul, S., Preda, M., Cobena, G., 2003. Adaptive On-line Page Importance Computation. Proc. 12th Int. Conf. on World Wide Web, p.280-290. [doi:10.1145/775152.775192]

Aggarwal, C.C., 2002. Collaborative Crawling: Mining User Experiences for Topical Resource Discovery. Proc. 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.423-428. [doi:10.1145/775047.775108]

Aggarwal, C.C., Al-Garawi, F., Yu, P.S., 2001. Intelligent Crawling on the World Wide Web with Arbitrary Predicates. Proc. 10th Int. Conf. on World Wide Web, p.96-105. [doi:10.1145/371920.371955]

Almpanidis, G., Kotropoulos, C., Pitas, I., 2005. Focused crawling using latent semantic indexing—an application for vertical search engines. *LNCS*, **3652**:402-413. [doi:10.1007/11551362_36]

Bharat, K., Henzinger, M.R., 1998. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. Proc. 21st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.104-111. [doi:10.1145/290941.290972]

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, **2**(2):121-167.

Calado, P., Cristo, M., Moura, E., Ziviani, N., Ribeiro-Neto, B., Gonçalves, M.A., 2003. Combining Link-based and Content-based Methods for Web Document Classification. Proc. 12th Int. Conf. on Information and Knowledge Management, p.394-401. [doi:10.1145/956863.956938]

Chakrabarti, S., van den Berg, M., Dom, B., 1999. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks*, **31**(11-16):1623-1640. [doi:10.1016/S1389-1286(99)00052-3]

Chau, M., Chen, H., 2003. Comparison of three vertical search spiders. *Computer*, **36**(5):56-62. [doi:10.1109/MC.2003.1198237]

Cho, J., Garcia-Molina, H., Page, L., 1998. Efficient Crawling through URL Ordering. Proc. 7th Int. Conf. on World Wide Web, p.161-172.

Davison, B.D., 2002. Topical Locality in the Web. Proc. 23rd Annual Int. ACM SIGIR Conf., p.272-279. [doi:10.1145/345508.345597]

Diligenti, M., Coetzee, F.M., Lawrence, S., Giles, C.L., Gori, M., 2000. Focused Crawling Using Context Graphs. Proc. 26th Int. Conf. on Very Large Databases (VLDB), p.527-534.

Elkan, C., 1997. Boosting and Naive Bayesian Learning. Technical Report No. CS97-557, Department of Computer Science and Engineering, University of California, San Diego.

Guan, Z., Wang, C., Chen, C., Bu, J., Wang, J., 2008. Guide Focused Crawler Efficiently and Effectively Using On-line Topical Importance Estimation. Proc. 31st Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval, p.757-758. [doi:10.1145/1390334.1390488]

Haveliwala, T.H., 2002. Topic-sensitive PageRank. Proc. 11th Int. Conf. on World Wide Web, p.517-526. [doi:10.1145/511446.511513]

Jain, A.K., Mao, J., Mohiuddin, K.M., 1996. Artificial neural networks: a tutorial. *Computer*, **29**(3):31-44. [doi:10.1109/2.485891]

Jamali, M., Sayyadi, H., Hariri, B.B., Abolhassani, H., 2006. A Method for Focused Crawling Using Combination of Link Structure and Content Similarity. IEEE/WIC/ACM Int. Conf. on Web Intelligence, p.753-756. [doi:10.1109/WI.2006.19]

Kleinberg, J., 1998. Authoritative Sources in a Hyperlinked Environment. Proc. 9th Annual ACM-SIAM Symp. on Discrete Algorithms, p.668-677.

Menczer, F., 1997. ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery. Proc. 14th Int. Conf. on Machine Learning, p.227-235.

Menczer, F., Pant, G., Srinivasan, P., Ruiz, M.E., 2001. Evaluating Topic-driven Web Crawlers. Proc. 24th Annual Int. ACM SIGIR Conf., p.241-249. [doi:10.1145/383952.383995]

Page, L., Brin, S., Motwani, R., Winograd, T., 1998. The Pagerank Citation Algorithm: Bringing Order to the Web. Technical Report, Stanford Digital Library Technologies, Stanford InfoLab.

Pant, G., Srinivasan, P., 2005. Learning to crawl: comparing classification schemes. *ACM Trans. Inf. Syst.*, **23**(4):430-462. [doi:10.1145/1095872.1095875]

Pant, G., Srinivasan, P., 2006. Link contexts in classifier-guided topical crawlers. *IEEE Trans. Knowl. Data Eng.*, **18**(1):107-122. [doi:10.1109/TKDE.2006.12]

Pant, G., Srinivasan, P., Menczer, F., 2002. Exploration versus Exploitation in Topic Driven Crawlers. Proc. 11th World Wide Web Workshop on Web Dynamics, p.1-10.

Rennie, J., McCallum, A., 1999. Using Reinforcement Learning to Spider the Web Efficiently. Proc. 16th Int. Conf. on Machine Learning, p.335-343.

Silva, I., Ribeiro-Neto, B., Calado, P., Moura, E., Ziviani, N., 2000. Link-based and Content-based Evidential Information in a Belief Network Model. Proc. 23rd Annual Int. ACM SIGIR Conf., p.96-103. [doi:10.1145/345508.345554]

Srinivasan, P., Menczer, F., Pant, G., 2005. A general evaluation framework for topical crawlers. *Inf. Retr.*, **8**(3):417-447. [doi:10.1007/s10791-005-6993-5]

Tang, T.T., Hawking, D., Craswell, N., Griffiths, K., 2005. Focused Crawling for Both Topical Relevance and Quality of Medical Information. Proc. 14th ACM Int. Conf. on Information and Knowledge Management, p.147-154. [doi:10.1145/1099554.1099583]

Yang, K., 2001. Combining Text- and Link-based Retrieval Methods for Web IR. Proc. 10th Text Retrieval Conf., p.609-618.