The Journal of Systems and Software 86 (2013) 187–197

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Semantic ranking of web pages based on formal concept analysis

YaJun Du*, YuFeng Hai

School of Mathematical and Computers Science, Xihua University, Chengdu 610039, Sichuan, China

ARTICLE INFO

Article history:
Received 10 November 2011
Received in revised form 14 April 2012
Accepted 15 July 2012
Available online 31 July 2012

Keywords:
Web crawler
Crawling direction
Search engine
Formal concept analysis

ABSTRACT

A web crawler is an important research component in a search engine. In this paper, a new method for measuring the similarity of formal concept analysis (FCA) concepts and a new notion of a web page's rank are proposed that use an information content approach based on users' web logs. First, an extension similarity and an intension similarity that analyze a user's browsing pattern and their hyperlinks are proposed. Second, the information content similarity between two nouns is computed automatically by examining their ISA and Part-Of hierarchy and using a user's web log. A method for computing the semantic similarity between two concepts in two different concept lattices (the base concept lattice and the current concept lattice) and finding the semantic ranking of web pages is proposed. Last, our experiment demonstrates that our crawler is more suitable for crawling focused web pages. It proves that the semantic ranking of web pages is useful and efficient for making a web crawler's choice of a web page for continuing work.

1. Introduction

A typical web search engine includes web crawling engine, web indexing engine, and web searching engine. The crawling engine finds new web pages and updates web pages existing in the database of the web search engine (Konstantin et al., 2011). A web crawler is designed to automatically download web pages by following hyperlinks in the Web and saving web pages to a local computer. In its period of immaturity, the program was also called wanderer, spider or web robot. Web spider is widely applied to search engines, e.g., AltaVista, DirectHit, Excite, Google, HotBot, Lycos and Yahoo. The workflow of a web crawler can be described roughly as follows (Cho et al., 1998; Du et al., 2005):

(1) A search engine assigns some URLs as the initial URLs for every web crawler. Then, the web crawler pushes them into a URL queue (queued URLs) in which each one instructs the web crawler where to travel in the Web.
(2) The web crawler starts working with the initial URLs.
(3) When the web crawler retrieves web pages, it extracts all of the URLs (current URLs) in the web pages.
(4) The web crawler adds them to the queued URLs.
(5) Where after, to continue crawling, the web crawler makes a choice of URLs from the queues URLs and deletes these crawled URLs.
(6) The web crawler repeats (2) to (5) until no URLs remain in the queue of URLs.

Coverage, relevance and precision (Sufyan Beg, 2005) are three indices that measure the performance of different web crawlers. Apparently, the performance of a web crawler is decided by steps (3), (4) and (5). Some web crawlers attempt to cover the most web pages possible; some attempt to crawl the most professional web pages, and some attempt to crawl the most accurate web pages for a user query. On the other hand, some web crawlers attempt to spend less time crawling web pages in specialized domains. No matter what type of web crawlers, their performance needs to be improved. In implementing these tasks, deciding how to select URLs from the queue of URLs and determining how to make a choice of URLs for next step pose two important challenges, and become two important research problems. In contemporary search engines, there are three types of user queries: specific queries, broad-topic queries and similar-page queries. To crawl and retrieve web pages efficiently, there are three typical methods for solving the above mentioned two challenges and problems: hyperlink-based, content-based, hyperlink-content-based methods (Rungsawang and Angkawattanawit, 2005). Aimed at the different user queries, each method has been developed into different models.

1.1. Hyperlink-based ranking

In the Internet, hyperlinked web pages are abstracted as directed graphs. The initial URLs make web crawlers travel around the Web and retrieve useful web pages. For hyperlink structural analysis, Page Rank and HITS (hyperlink induced topic search) are two

* Corresponding author. Tel.: +86 28 87720554.
E-mail address: duyajun@mail.xhu.edu.cn (Y. Du).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2012.07.040

popular algorithms; they have been adopted for web crawlers of search engines such as Google, HotBot, etc. In addition, back-link, forward-link, etc. are also used to crawl the Web. PageRank (Brin and Page, 1998) is a famous algorithm that was originally designed for ranking web pages of search results in Google. A hyperlink from web page A to web page B may indicate that web page A is related to web page B, or that web page A is recommending, citing or endorsing web page B. Afterward, Cho et al. (1998) deemed that it is important for a web crawler to visit more important web pages first. A web PageRank, PR, is used to measure the importance of a web page in the process of visiting a web page. The importance of web page A is computed by Eq. (1). The PageRank evaluates the importance of web page A from a hyperlink viewpoint. The PageRank algorithm was designed on behalf of all of the authors of web page linking to page A, but not for one authors' specific interest. Therefore it is suitable for searching broad search topics. We denote the web crawlers that choose a URL from the queue of URLs for the next crawling phase by using a breath-first algorithm and the PageRank of the URL's corresponding web page as breath-first web crawlers (***BFCrawlers***).

$$PR(A) = (1-d) + d * \left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_k)}{C(T_k)} \right). \quad (1)$$

Where $PR(T_i)$ is the PR value of web page T_i ($i = 1, 2, \ldots, k$), which links to web page A, and $C(T_i)$ is defined as the number of hyperlinks that leave web page T_i ($i = 1, 2, \ldots, k$). The PR of web page A is not related to the initial PR of $T_1, T_2, \ldots, T_k$. The term d is an attenuation factor, often equal to 0.85 for the process of computing a PageRank.

Kleinberg (1999) proposed HITS. HITS is a link-based model for broad search topics, and the web pages in the web graph induced by their hyperlink structure are defined to authorities and hubs. For a given broad-topic query σ, the authoritative web pages are determined from the web graph by an analysis of two operations ($\mathcal{I}$ and $\mathcal{O}$) of the hyperlink structure, which requires the computation of the authority weight $x^{\langle p \rangle}$ and hub weight $y^{\langle p \rangle}$ of web page $p \in V$. The operations $\mathcal{I}$ and $\mathcal{O}$ are defined as follows.

$$\mathcal{I} : x_{i+1}^{\langle p \rangle} \leftarrow \sum_{q:(q,p)\in E} y_i^{\langle q \rangle}, \qquad \mathcal{O} : y_{i+1}^{\langle p \rangle} \leftarrow \sum_{q:(p,q)\in E} x_i^{\langle q \rangle}.$$

Here i is the number of iterations, and $x^{\langle p \rangle}$ and $y^{\langle p \rangle}$ are normalized so that their squares sum to 1:

$$\sum_{p\in V} (x^{\langle p \rangle})^2 = 1, \sum_{p\in V} (y^{\langle p \rangle})^2 = 1.$$

However, in real applications, the HITS algorithm produces some problems, such as the time and space costs of constructing the subgraph of the search topic are high; $G = (V, E)$. V is the web page set, and E is the hyperlink structure among web pages. HITS is also not suitable for specific queries. Some improved algorithms such as the probabilistic analogue to HITS (Cohn and Chang, 2000), hub-averaging-Kleinberg and threshold-Kleinberg (Allan and Roberts, 2001), and the stochastic approach for link-structure analysis (Lempel and Moran, 2000) have been developed.

1.2. Content-based ranking

Within the development of web information retrieving technology, page rank and HITS are only suitable to crawl web pages for broad-topic queries. However, focused web crawlers efficiently retrieve web pages for specific queries and similarity page queries. Content-based ranking technologies were developed for this purpose. During web crawler crawls, the similarity between a user query and a web page is used to measure the rank of a web page. To compute this similarity, web page p and user query q are formalized as two n-dimensional vectors ($w_{p1}, \ldots, w_{pn}$ and $w_{q1}, \ldots, w_{qn}$, respectively). The terms w_{pi} and w_{qi} are the frequency scores of the ith words in the web page p and user query q, respectively. In the Boolean Model, if the ith word appears in p or q, then w_{pi} or $w_{qi} = 1$; otherwise, w_{pi} or $w_{qi} = 0$.

In the vector space model, term frequency-inverse document frequency (TF-IDF) is used as a classical global weighting method for the two vectors, and the TF-IDF score of each word in a the web page or user query measures the importance of the word in the vector. If the ith word appears in web page p and user query q, w_{pi} and w_{qi} are weighted proportionally to the term frequency and inversely proportional to the document frequency (Salton et al., 1975; Cohn and Chang, 2000). In a more sophisticated extension of this approach, the documents of a collection are ranked in a decreasing order of their probability of relevance to a users query, a method known as probability ranking (Jones, 1979; Almpanidis et al., 2007). The w_{ij} of the ith word t_i in jth web page p_j or jth user query is measured by Eq. (2). The similarity between a user query and a web page is computed by the Cosine distance formula, and this similarity evaluates the importance of a web page. We denote the web crawlers that choose URLs from the queue of URLs for the next crawling phase by using the cosine distance of the URL's corresponding web page as Cosine Similarly Focused Web Crawlers (***CosFCrawler***).

$$w_{ij} = \frac{cfw(t_i) * tf(t_i, d_j) * k + 1}{k * ((1-b) + b * N(D_j))) + tf(t_i, d_j)}. \quad (2)$$

Where the collection frequency weight ($cfw(t_i) = \log^{(n/n_i)}$) of term t_i implies that terms appearing in few documents are more valuable than those appearing in many (Almpanidis et al., 2007). The term n is the number of documents in the collection, and n_i is the occurrence of word t_i in the collection. The variable $tf(t_i, d_j)$ is the term frequency weight defined as the number of occurrences of term t_i in the document d_j or user query d_j. $N(D_j) = DL(d_j)/\overline{DL}$; where, $DL(d_j)$ is the document length of document d_j, and $\overline{DL}$ is the average document length in the collection of n web pages. The terms b and k are adjustment factors.

Park et al. (2011) proposed an ontology selection and ranking model consisting of selection standards and metrics based on better semantic matching capabilities which can enhance the ontology selection and ranking method practically and effectively by enabling semantic matching of taxonomy or relational linkage between concepts. The ranking can identify measures-rank ontologies in the given context and weight assigned to each selection measure.

1.3. Hyperlink-content-based ranking

A meaningful and interesting discussion encompasses finding an appropriate balance between the criteria of relevant and popular web pages. A new focused web crawler based on a Hidden Markov Model has been proposed (Liu et al., 2006). During the crawling phase, the web crawler associates a priority value with each URL, and the URLs with the higher priority value are added to the queue of URLs. The priority value of a URL is computed by a combination of its hyperlinks and the content of its web page. The visited web pages are collected and clustered. Whereas, web page sequences target pages that are extracted from the hyperlink structure among web pages from different clusters by using a Hidden Markov Model. We denote the web crawlers that choose web pages for the next round of crawling by using a Hidden Markov Model as Hidden Markov Model Focused Web Crawlers.

Diligenti et al. (2000) proposed that the link context graph of the web pages that have been visited can be used as the context information for later crawling and put to use in a focused web crawling system. In order that the URLs of a relevant topic can be captured

Table 1
Differences among three rankings.

	Hyperlink	Content	Semantic meanings
Hyperlink-based ranking	Y	N	N
Content-based ranking	N	Y	Y
Hyperlink-content-based	Y	Y	Y

to access a desired URL, this model finds the goal URL by searching the parent URLs of the inner-layer (the links that go to URLs inside them) and forming a connection-layered graph of the link relationships, layer by layer. The web crawlers that use a link context graph can be divided into two phases. (1) In the learning phase, a link context graph and a classifier of web pages in a seed set should be built for users. (2) In the crawling phase, the main job is to guide the focused web crawling with the classifier. After learning a user's topic information, the classifier will start the focused web crawling. We denote the web crawlers that choose web pages for the next round of crawling by using a link context graph as Hidden Link Context Graph Focused Web Crawlers (***LCGCrawlers***).

To prevent a web crawler from accessing irrelevant URLs and allow the crawler to harvest as many relevant URLs as possible, Hsu and Wu (2006) proposed a method based on a Relevancy Context Graph. It is taken for granted that there must be a position for each URL on the Web. The URLs of the same topic could be collected to form a relevancy context graph. In this relevancy context graph, each URL that is related to the topic can be linked to each other to some extent by some kind of semantic relationship. Hsu et al. constructed their relevancy context graph by forming each layer in an order of decreasingly similarity to target URLs. They assigned each layer an invariable similarity value; the similarity value of layer i is α^i. The core layer is supposed to be a 0-layer, and its similarity value is 1, and by analog, from inside to outside. Besides, the value α can be adjusted. We denote the web crawlers that choose web pages for the next crawling phase by using a relevancy context graph as Hidden Relevancy Context Graph Focused Web Crawlers (***RCGCrawlers***).

By researching the characteristics of current focused web crawling strategies, Du and Dong (2009), Yang et al. (2008) brought forward a top-specific crawling strategy that uses measurements of the formal concept context graph. By matching unvisited web pages with a concept lattice of the user's topic interest, the rankings of unvisited web pages are computed to pick out relevant hyperlink. Zheng et al. (2008) proposed a learnable focused crawling framework based on ontology. An ANN (artificial neural network) was constructed using a domain-specific ontology that utilizes manually predefined concept weights to calculate the relevance scores of web pages. Jung (2009) focused on domain ontology and business rules. They exploited the context-based focused crawler architecture to discover local knowledge from interlinked systems. Batsakis et al. (2009) proposed several variants of state-of-the-art crawlers that rely on web page content and link information to estimate the relevance of web pages to a given topic. In order to benefit both user requests and domain semantics, Yang (2010) proposed ontology-supported website models and develop a focused crawler-OntoCrawler which can provide a semantic level solution for an information agent.

1.4. Contributions and outline of this paper

Three factors are adopted to describe the differences among Hyperlink-Based Ranking, Content-Based Ranking and Hyperlink-Content-Based rankings of URLs (Table 1): the hyperlink relation among web pages, the content (key words) and semantic meaning of web pages. The hyperlink-based ranking only considers hyperlinks among web pages. And so, the PageRank and HITS algorithms lack two important factors (the content of web pages and semantic meanings among web pages) for the ranking of URLs. The content-based ranking only considers the content correlation among web pages by analogizing key words, and it takes on the part semantic meanings. The hyperlink-content-based ranking not only considers the hyperlinks but also the part semantic meanings among the web pages. From semantic perspective of web pages, some algorithms for the ranking of URLs have been proposed. However, the algorithms for computing the rankings of URLs, such as Markov model (Liu et al., 2006), link context graph (Diligenti et al., 2000), relevancy context graph (Hsu and Wu, 2006), domain-specific ontology (Zheng et al., 2008), etc. only consider the key words and their frequencies in web pages, and part semantic meanings of web pages. They lack analyzing the concepts of web pages, the rankings of URLs lack the semantic meanings with reality significance. In this paper, the semantic ranking of URLs with the reality significance is proposed.

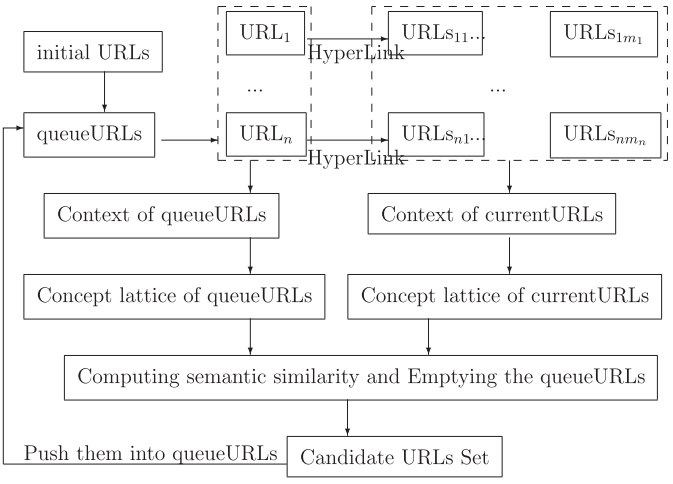

Fig. 1. Idea of a semantical ranking system.

To retrieve web pages that are all satisfactory to users, our web crawler (Fig. 1) works as follows:

(1) Initial URLs of a user query are selected by famous search engines, such as Google, AltaVista or HotBot. After submitting a user query, the web crawler expands the query, allocates some key words to these famous search engines by their performance, and selects initial URLs by comparing their search result.
(2) These URLs are used as the queue of URLs and pushed them into the queue. Their URLs are extracted as the current URLs from which they are linked. These form the contexts of the queue of URLs and the current URLs, respectively.
(3) All of the concepts of the queued URLs and current URLs are extracted. Then, two concept lattices for these concepts are constructed.
(4) The semantic similarity between each concept of a queued URLs and each concept of a current URL is computed. At last, the semantic ranking of the URLs of web pages in the current URLs is computed.
(5) The queue of URLs is emptied, and URLs of some web pages with greater similarity to the search topic are chosen from the current URLs and pushed into the queue of URLs. Then, steps (2) and on are repeated until the queue of URLs is NULL.

In Sections 2 and 3, we give the definition of the Concept Lattice of URLs, where concept extents are represented by sets of URLs and concept intents are represented by sets of user keywords. We

propose a novel similarity measure that uses FCA and ontology to support web crawlers.

2. Concept lattice of URLs

The concept lattice was systematically built up by Wille (1982, 1989, 1992). A concept lattice and its corresponding hasse diagram reflect a conceptual hierarchy. It is constructed on formal context. In this section, we discuss the notions of the formal context, formal concept and concept lattice of initial URLs and current URLs.

Definition 1. A formal context of URLs is a tripe $T = (\text{URLs}, W, Q)$, where each URL $\in$ URLs is interpreted to an object; each w $\in$ W is interpreted to an attribute, and W is the common key word set of web pages identified by URLs. The URLs and W cannot be empty sets; $Q \subseteq \text{URLs} \times W$ is a binary relation set. If $(\text{URL}, w) \in Q$, then it means that an object URL has the attribute w.

When a user submits his query to a search engine, this query reflects a concept related to user knowledge. In natural language, a noun represents a concept. However, the essence of a concept can be formalized into objects and attributes. For example, student is a concept; student implies a person who studies in school, and some attributes include the student's name, student-id, etc. In URLs, some web pages also form concepts related to a user query. A concept of URLs can be formalized into a duality (objects, attributes); it reflects that these objects in duality take on common attributes. These attributes are only shared by these objects. To introduce the definition of a concept of URLs, we rewrite two set-valued functions: $\uparrow$ and $\downarrow$(Wille, 1982),

$$\uparrow: \quad P(\text{URLs}) \to P(W), X^{\uparrow} = \{w | w \in W; \forall \text{URL} \in X, (\text{URL}, w) \in Q\},$$

$$\downarrow: \quad P(W) \to P(\text{URLs}), Y^{\downarrow} = \{\text{URL} | \text{URL} \in \text{URLs}; \forall w \in Y, (\text{URL}, w) \in Q\}.$$

Definition 2. A concept of URLs is a duality $(X, Y) \in P(\text{URLs}) \times P(W)$ such that $X^{\uparrow} = Y$ and $Y^{\downarrow} = X$. The set X is called the extension of the concept. The set Y is the intension of the concept.

The greatest concept I and smallest concept O of URLs can be described, respectively, as follows:

$$\bigvee_{i=1}^{n}(X_i, Y_i) = ((\bigcup_{i=1}^{n} X_i)^{\uparrow\downarrow}, \bigcap_{i=1}^{n} Y_i), \bigwedge_{i=1}^{n}(X_i, Y_i) = (\bigcap_{i=1}^{n} X_i, (\bigcup_{i=1}^{n} Y_i)^{\downarrow\uparrow}). \quad (3)$$

Given two concepts (X_1, Y_1) and (X_2, Y_2) of URLs, $(X_1, Y_1) \leq (X_2, Y_2)$ if and only if $X_1 \subseteq X_2$ (or equivalently $Y_1 \supseteq Y_2$).

Definition 3. Let C be all concepts of URLs, and $L(T)$ be $(C, O, I, \leq)$. We denote $L(T)$ as a concept lattice of URLs.

Definition 4. (The base concept lattice (BL) and the current concept lattice (CL)). Consider two formal contexts $T_1 = (\text{queueURLs}, W_1, Q_1)$ and $T_2 = (\text{currentURLs}, W_1, Q_1)$. They generate two concept lattices BL and CL for T_1 and T_2, respectively.

Example 5. A user query including two key words Internet, Spider submitted to PISE (Personalized Intelligence Search Engine (Du and Li, 2010)). PISE selects 3 initial URLs (represented as U_1, U_2, U_3) as the queue of URLs and 5 URLs (represented as U_1, U_2, U_3, U_4, U_5) as the current URLs in the first iteration of crawling. Internet, Spider and Ontology are the key words shared by the web pages of the queued URLs and chosen from these web pages. Internet, Technology, Network, Web page, Information and Spider are the key words shared by the web pages of the current URLs and chosen from these web pages. Table 2 (top) is the formal context of the queued URLs. Table 2 (bottom) is the formal context of the current URLs. From these two formal contexts, we extract some concepts as follows (Fig. 2):

Table 2
Two formal contexts for the user query: Internet, Spider.

	Internet	Spider	Ontology
U_1	√	√	
U_2		√	√
U_3	√	√	√

	Internet	Technology	Network	Web page	Information	Spider
U_4				√	√	√
U_5	√	√	√	√		√
U_6	√	√	√	√		
U_7		√	√		√	√
U_8		√		√	√	√

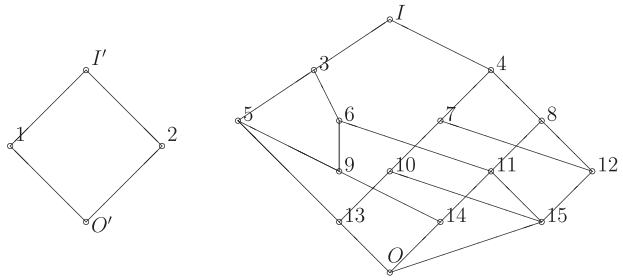

Fig. 2. Concept lattice of queued URLs and current URLs for the user query Internet, Spider.

1. $(\{U_1, U_3\}, \{\text{Internet, Spider}\})$;
2. $(\{U_2, U_3\}, \{\text{Ontology, Spider}\})$;
   $I'.(\{U_1, U_2, U_3\}, \text{Spider})$;
   $O'.(\{U_3\}, \{\text{Internet, Spider, Ontology}\})$;
3. $(\{U_4, U_5, U_6, U_8\}, \{\text{Web page}\})$;
4. $(\{U_5, U_6, U_7, U_8\}, \{\text{Technology}\})$;
5. $(\{U_4, U_7, U_8\}, \{\text{Information, Spider}\})$;
6. $(\{U_4, U_5, U_8\}, \{\text{Web page, Spider}\})$;
7. $(\{U_5, U_6, U_7\}, \{\text{Technology, Network}\})$;
8. $(\{U_5, U_6, U_8\}, \{\text{Technology, Web page}\})$;
9. $(\{U_4\}, \{\text{Web page, Information, Spider}\})$;
10. $(\{U_5, U_7\}, \{\text{Technology, Network, Spider}\})$;
11. $(\{U_5, U_8\}, \{\text{Technology, Web page, Spider}\})$;
12. $(\{U_5, U_6\}, \{\text{Internet, Technology, Network, Web page}\})$;
13. $(\{U_7\}, \{\text{Technology, Network, Information, Spider}\})$;
14. $(\{U_8\}, \{\text{Technology, Web page, Information, Spider}\})$;
15. $(\{U_5\}, \{\text{Internet, Technology, Network, Web page, Spider}\})$;
    $I.(\{U_4, U_5, U_6, U_7, U_8\}, \Phi)$;
    $O.(\Phi, \{\text{Internet, Technology, Network, Web page, Information, Spider}\})$;

3. Selection of a URL for the next crawling iteration

In this section, we propose an extension similarity and an intension similarity based on a user's browsing pattern. Last, we develop the semantic ranks of web pages.

3.1. Constructing a domain ontology with a user's browsing pattern

Although domain ontology and FCA have different purposes, they offer tools for modeling concepts (Formica, 2006, 2008). In real world applications, a concept includes its extensional and intensional aspects. The extension of a concept is all of the objects in which each object takes on all attributes of the concept, while the intension of the concept is all of the attributes in which each attribute is shared commonly by all objects of the concept. Given a formal context (domain), FCA works to formalize a domain of user interest by a formal pair (objects, attributes). Its concept gives an exact description for a realistic concept. Ontology emphasizes the intensional component to model the domain interest (Uschold and

Gruninger, 1996). A domain ontology is a "formal, explicit specification of a shared conceptualization" (Ding et al., 2002). A domain ontology contains a set of interrelated concepts, each associated with a formal definition that provides an unambiguous meaning of the concept in the given domain (Formica, 2006, 2008). Therefore, a domain ontology should be explained as a set of concepts and the relationships among them evaluated by a panel of experts in the given domain.

Bain (2003) has introduced approaches for combining FCA and ontology in many applications, such as the semantic Web, information retrieval, etc. The semantic similarity of two concepts ($C_1 = (X_1, Y_1)$ and $C_2 = (X_2, Y_2)$), based on the notion of Formica (2006, 2008), includes not only the extensions of the two concepts but also their intentions. Formica (2006) considered a domain ontology ϑ and defined the concept similarity (Sim) of two concepts C_1 and C_2 of the different concept lattices. Let $n = |Y_1|$, $m = |Y_2|$, $r = max(|X_1|, |X_2|)$ and suppose that $n \le m$. The set $\mathcal{P}(Y_1, Y_2)$ is defined by all possible sets of n pairs of attributes $\mathcal{P}(Y_1, Y_2) = \{\{\langle a_1, b_1\rangle, \ldots, \langle a_n, b_n\rangle\} | a_i \in Y_1, b_i \in Y_2, \forall i = 1, \ldots, n, and a_i \neq a_k, b_i \neq b_l, \forall k, l \neq i\}$. $\forall P \in \mathcal{P}(Y_1, Y_2)$, $\langle a, b\rangle \in P$, we denote $\langle a, b\rangle$ to a concept descriptor. $Sim(C_1, C_2)$ is defined as follows:

$$Sim(C_1, C_2) = \frac{|X_1 \cap X_2|}{r} * w + \frac{1}{m} \max_{P \in \mathcal{P}(Y_1, Y_2)} \sum_{\langle a,b\rangle \in P} as(a, b) * (1 - w). \quad (4)$$

In Eq. (4), $|X_1 \cap X_2|/r$ and $(1/m) \max_{P \in \mathcal{P}(Y_1, Y_2)} \sum_{\langle a,b\rangle \in P} as(a, b)$ are the extension similarity and intension similarity of two concepts, respectively. Here $as(a, b)$ is the similarity between word a and word b, $w \in [0, 1]$ is a weight given by the user to adjust the proportions of the extension similarity and intension similarity of two concepts. In this paper, we improve the extension similarity and intension similarity of two concepts.

3.2. Extension similarity

We consider that a hyperlink and click between web page A and web page B are two very important parts of their semantic relationship. In the Web, a hyperlink from web page A to web page B suggests that "Web page A and B might be on same topic" or "the author of Web page A recommend Web page B to the user" (Brin and Page, 1998). It implies an axiomatic semantic relationship between web pages A and B. PageRanks of web pages reflect the evaluations (out-degrees and in-degrees of web pages) of authors. These authors are the experts of the different domains. On the other hand, a click from web page A to web page B could suggest that "the user approve that web pages A and B might be on same topic". A click also implies an axiomatic semantic relationship between web pages A and B. It indicates that the intension of concepts (topics) of web page A is similar to those of web page B. The user web log of a search engine records the abundant history click-data of users. In fact, the knowledge for a group of users with the same interests is very outstanding knowledge for the special domain; users are the finest experts. When considering two concepts ($(X_1, Y_1) \in BL$ and $(X_2, Y_2) \in CL$), the hyperlinks between URLs in X_1 and URLs in X_2 reflect the semantic relationship of their extensions; the click-data between URLs in X_1 and URLs in X_2 reflect the semantic relationship of their extensions too. Although they cannot share objects between the concepts of BL and CL, we note that there exist some hyperlinks and click-data that are shared among URLs in BL and CL. They reflect the semantic relationship of the extensions of the two concepts in BL and CL.

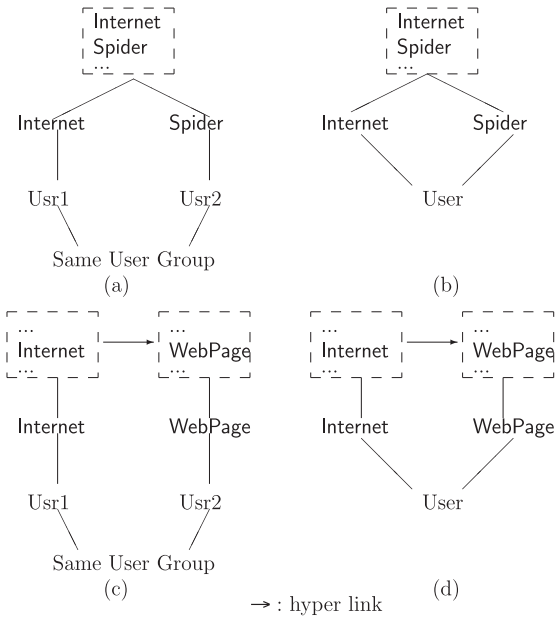

Fig. 3. ISA (Internet, Spider) and Part-Of (Internet, Web Page) identified from the clicked Web page that includes the user query.

Definition 6. (Extension similarity) Consider two concept lattices BL and CL, $(X_1, Y_1) \in BL$ and $(X_2, Y_2) \in CL$. Let $Sim_{\text{extension}}$ be the extension similarity between (X_1, Y_1) and (X_2, Y_2).

$$Sim_{\text{extension}} = \frac{|X_1 \rightarrow_L X_2| + |X_1 \leftarrow_L X_2| + |X_1 \rightarrow_C X_2| + |X_1 \leftarrow_C X_2|}{4 * max(|X_1|, |X_2|)}.$$

Where $|X_1 \rightarrow_L X_2|$ and $|X_1 \rightarrow_C X_2|$ are the number of hyperlinks in which URLs in BL link to URLs in CL and the number of clicks from URLs of CL to URLs in BL, respectively. $|X_1 \leftarrow_L X_2|$ and $|X_1 \leftarrow_C X_2|$ are the number of hyperlinks in which URLs in CL link to URLs in BL and the number of clicks from URLs of BL to URLs in CL, respectively.

3.3. Intension similarity

To compute the semantic similarity of the intensions of two concepts in a given domain ontology, the similarity for any pair of concept descriptors should be considered. Formica (2008) replaces this similarity with information content similarity scores that can be automatically computed by any lexical database, such as Wordnet, Conceptnet and Cyc (Conesa et al., 2008). In a lexical database for English nouns, the relationships among nouns such as their ISA, Part-Of, etc. are appointed by some linguists or other specialists for the given domain. Because the computation relies on domain expertise, it is inconvenient to program for a real application. However, for the personalized web crawler of our PISE, which retrieve timely web pages from the Internet for user queries, when users browse these web pages, a great deal of history knowledge is saved. So, a domain ontology that determines the semantic relationship of two concepts of CL and BL should rely on the history knowledge of the user. On the other hand, because the user web log of our PISE offers abundant history information, we make full use of this information to measure the similarity of concept descriptors. To compute the information content similarity, we determine the semantic relationships among nouns as follows:

- If the different users of the same user group submit two different key words (n_1, n_2), then click the same web pages, the two key words own the semantic relationship ISA(n_1, n_2).

Example 7. In Fig. 3(a), user1 and user2 are classified in the same interest group. User1 submits the keyword n_1=Internet to PISE, and user2 submits keyword n_2=Spider to PISE. Both user1 and user2 clicked the same web page having the keywords Internet and Spider. So, we consider that the ISA(Internet, Spider) holds.

- If a user submits two different key words (n_1, n_2) and clicks the same web pages for both, then the two key words own the semantic relationship ISA(n_1, n_2).

  Example 8. In Fig. 3(b), a user submits the keyword n_1=Internet to PISE and also submits the keyword n_2=Spider to PISE. He clicks a web page including the keywords Internet and Spider for both. So, we consider that the ISA(Internet, Spider) holds.

- If different users of the same user group submit two different key words (n_1, n_2) and they click web pages in which there exist hyperlinks from web page with the key word n_1 to the web page with the key word n_2, then the two key words own the semantic relationship Part-Of(n_1, n_2). This indicates that n_2's meaning is a part of n_1.

  Example 9. In Fig. 3(c), user1 and user2 are classified into the same interest group. User1 submits the keyword n_1=Internet to PISE, and user2 submits the keyword n_2=Web page to PISE. Although user1 and user2 did not click on the same web page including the keywords Internet and Web page, the web page with the keyword Internet clicked by user1 has a hyperlink to the web page with the keyword Web page clicked by user2. So, we consider the Part-Of(Internet, Web page) holds.

- If a user submits two different key words (n_1, n_2) and clicks on web pages for which there exist hyperlinks from the web page with the key word n_1 to web page with the key word n_2, then the two key words own the semantic relationship Part-Of(n_1, n_2) (Fig. 3(d)). This indicates that n_2's meaning is a part of n_1.

  Example 10. In Fig. 3(d), a user submits the keyword n_1=Internet to PISE, and also submits the keyword n_2=Web page to PISE. Although the user did not click on a web page including the keywords Internet and Web page, but the user clicked a web page including the keywords Internet has a hyperlink to web page clicked that has the keywords Web page. So, we consider that Part-Of(Internet, Web page) holds.

- According to the four cases above, if the relationship of two key words n_1, n_2 is ISA(n_1, n_2) and Part-Of(n_1, n_2), then we designate their relationship as ISA(n_1, n_2).

We must note that Part-Of(n_1, n_2) indicates that key word n_2's semantic meanings also includes in key word n_1's semantic meanings.

Definition 11. (Lexical nouns database for *BL* and *CL*). A lexical database (Ω) for *BL* and *CL* is 4-tuple (N_{BL}, N_{CL}, $f(N)$, R), where N_{BL} and N_{CL} are the sets of key words in which each one is an attribute of the formal context of *BL* or *CL*, respectively. $N = N_{BL} \bigcup N_{CL}$, $f(N)$ is a function from N_{BL} or N_{CL} of the positive integers for which every value represents a click-number of a web page of a user web log after submitting the key word $n \in N$ to a web crawler. R is a set of relationships between N_{BL} and N_{CL} (such as ISA and Part-Of).

Example 12. Consider a user web log of PISE for the user query Internet Spider, N_{BL} = {Internet, Spider, Ontology}, N_{CL} = {Internet, Technology, Network, Web page, Information, Spider}. The part of the user web log for the user query Internet Spider is listed in Table 3; f(Internet) = 32510, ⋯, f(Network) = 43891. Obviously, R = {ISA(Spider, Spider), ISA(Internet, Internet), ISA(Internet, Spider), ISA(Technology, Network), ISA(Web page, Information), Part-Of(Internet, Technology), Part-Of(Internet, Network), Part-Of(Spider, Information), Part-Of(Spider, Web page), ⋯}.

Table 3
A fragment of a web log of the user query: Internet Spider.

Key word	Click-number	Click-sequence	Hyper links
Internet	3251	$U_1 \to {}_C U_7 \to {}_C \ldots$	$U_1 \to {}_L U_6$
Technology	2458	$U_3 \to {}_C U_4 \to {}_C \ldots$	$U_7 \to {}_L U_3$
Web page	12983	$U_2 \to {}_C U_6 \to {}_C \ldots$	$U_2 \to {}_L U_8$
Information	67856	$U_2 \to {}_C U_6 \to {}_C \ldots$	$U_2 \to {}_L U_8$
Spider	565	$U_1 \to {}_C U_8 \to {}_C \ldots$	$U_1 \to {}_L U_8$
Network	4389	$U_3 \to {}_C U_5 \to {}_C \ldots$	$U_3 \to U_5$
Ontology	...	...	...

The weighted hierarchy of the information content approach (Resnik, 1995; Lin, 1998) has not paid attention to the Part-Of relationship, but only paid attention to the ISA relationship. To compute the information content similarity, Formica (2008) focused on determining the ISA relationship. The notion of a weighted hierarchy is constructed on the probability of a noun n at every node. In this paper, we consider that the ISA and Part-Of relationships are important and frequent semantic relationships in a weighted hierarchy, this allows us to define the notion of a weighted ISA and Part-Of hierarchy.

Definition 13. (Weighted ISA and Part-Of hierarchy). Given a lexical database Ω for *BL* and *CL*, let ∂ be the ISA and Part-Of hierarchy; ∂ is a direct Graph. The nodes (key words) in the same layer reflect the ISA relationships among these key words. On the other hand, the nodes (key words) in different layers for which there exist connecting paths reflect the Part-Of relationships among these key words. Part-Of(n_1, n_2) can be represented as the directed edge $n_1 \to n_2$, iff $n_1 \in N_{BL}$ and $n_2 \in N_{CL}$. The probability of every key word is computed as:

$$p(n) = \frac{f(n)}{\sum_{n \in N_{BL} \bigcup N_{CL}} f(n)}.$$

In a user web log of our PISE, $\sum_{n \in N_{BL} \bigcup N_{CL}} f(n) = 87482$. The connecting weight $W_{ISA}(n_1, n_2)$ of two key words n_1, n_2 attached to them for ISA relationships is 1.0. The connecting weight $W_{\text{Part-Of}(n_1,n_2)}$ of two adjacent key words n_1, n_2 attached to them for Part-Of relationships is defined as:

$$W_{\text{Part-Of}}(n_1, n_2) = \frac{1}{Max(path(n_1), path(n_2)) + \alpha}.$$

Where $path(n)$ is the node number of the maximum path from the root to node n, and $\alpha \le 0.5$ is an adjustment factor for the weighted ISA and Part-Of hierarchy.

The weighted ISA and Part-Of hierarchy is a directed graph and has a unique top node (key word). The top node marked up "Top" or key words are determined as follows:

(1) If a the user submits a key word n and the key word has no other key words that they have an ISA relationship with key word n in the Lexical Nouns Database for the *BL* and *CL*, then the top node is the key word n.
(2) If a user submits a key word n and the key word has the other key words that they have an ISA relationship with key word n in the Lexical Nouns Database for the *BL* and *CL*, we assume that the ISA and Part-Of hierarchy has the top node (the most general key word, marked "Top"), the node "Top" retains a Part-Of relationship with the key word n, and these key words that maintain ISA relationships with key word n; let $p(Top) = max(p(n))$.
(3) If a user submits some key words and these key words have ISA relationships among themselves, then we assume that the

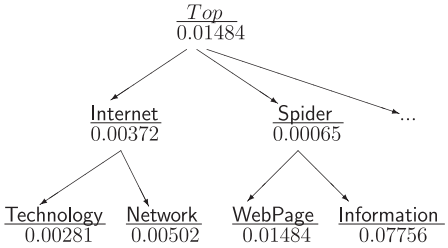

Fig. 4. A fragment of a weighted ISA and Part-Of hierarchy.

ISA and Part-Of hierarchy has the top node (the most general key word, marked "Top"). The node "Top" retain its Part-Of relationships with these key words; let $p(top) = max(p(n))$.

Example 14. A fragment of the weighted ISA and Part-Of hierarchy derived from a user web log of the user query Internet Spider (Table 3) is shown in Fig. 4. Table 3 indicates that, when the user submitted the key words Internet, Spider, he clicked the same web page of URL U_1. The ISA(Internet, Spider) holds, and the weighted ISA and Part-Of hierarchy owns a unique top node marked "Top". Every node in Fig. 4 is represented by a *Keyword*/p(*Keyword*). We let $p(TOP)$ be the minimum of all $p(n)$.

Definition 15. (Information content similarity (*ics*)). Consider a lexical database Ω for *BL* and *CL*, let ∂ be the weighted ISA and Part-Of hierarchy for the two key words $n_1 \in N_{BL}, n_2 \in N_{CL}$, To find ISA and Part-Of relationships of n_1 and n_2 to their shared node, we revise the information content similarity $ics(n_1, n_2)$ of n_1, n_2 as follows:

$$ics(n_1, n_2) = \begin{cases} 1.0, & \text{if} ISA(n_1, n_2), \\ \dfrac{2\log^{p(n^*)}}{\log^{p(n_1)} * w_1 + \log^{p(n_2)} * w_2}, & \text{if} \text{Part} - \text{Of}(n_1, n_2). \end{cases} \quad (5)$$

Where n^* is a key word providing the maximum information content shared by n_1, n_2. Let w_1 and w_2 be the weights of "from n^* to n_1" and "from n^* to n_2", respectively. $w_1 = \sum_{m,n\in N_1} W_{\text{Part-Of}}(m, n)$, $w_2 = \sum_{m,n\in N_2} W_{\text{Part-Of}}(m, n)$, N_1 and N_2 are the sets of key words in which each key word belongs to the nodes of the maximum path from n^* to n_1 and from n^* to n_2 in the weighted ISA and Part-Of hierarchy ∂, respectively. m is a key word that has a Part-Of relationship with n.

Example 16. Continuous Example 5: (1). Consider a case in which n_1=Network and n_2=Network. The maximum information content shared by n_1, n_2 is Network, and

$ics(\text{Network}, \text{Network}) = 1.$

(2). Consider a case in which n_1=Internet and n_2=Web Page. The maximum information content shared by n_1, n_2 is *Top*, and letting $\alpha = 0.2$,

$$ics(\text{Internet}, \text{WebPage}) = \frac{2 * \log^{p(Top)}}{\log^{p(\text{Internet})} * w_1 + \log^{p(\text{WebPage})} * w_2}$$

$$= \frac{2 * \log^{0.01484}}{\log^{0.00372} * (1/(1+0.2)) + \log^{0.01484} * ((1/(1+0.2)) + (1/(2+0.2)))} = 0.8351.$$

(3). Consider a case in which n_1 = Technology, and n_2 = Network. The maximum information content shared by n_1 and n_2 is Internet, and letting $\alpha = 0.2$,

$$ics(\text{Technology}, \text{Network}) = \frac{2 * \log^{p(\text{Internet})}}{\log^{p(\text{Technology})} * w_1 + \log^{p(\text{Network})} * w_2}$$

$$= \frac{2 * \log^{0.00372}}{\log^{0.00281} * ((1/(1+0.2)) + (1/(2+0.2))) + \log^{0.00502} * ((1/(1+0.2)) + (1/(2+0.2)))} = 0.77780. \quad (6)$$

From the above 3 cases, we conclude that the greater the distance from n_1, n_2 to the top node is, the lower the $ics(n_1, n_2)$ is.

Definition 17. (Intension similarity). Consider two concept lattices *BL* and *CL* for which $(X_1, Y_1) \in BL$ and $(X_2, Y_2) \in CL$. Let $Sim_{\text{intension}}$ be the intension similarity between (X_1, Y_1) and (X_2, Y_2).

$$Sim_{\text{intension}} = \frac{1}{|\mathcal{P}(Y_1, Y_2)|} \max_{P\in\mathcal{P}(Y_1,Y_2)} \Big(\sum_{\langle a,b\rangle\in P} ics(a, b) \Big).$$

3.4. The semantic ranking of web pages in concept lattice CL

Definition 18. Consider two concept lattices *BL* and *CL* for which $(X_1, Y_1) \in BL$ and $(X_2, Y_2) \in CL$. The concept similarity (*Sim*) between (X_1, Y_1) and (X_2, Y_2) includes two parts: the extension similarity and the intension similarity. We define it as follows:

$$Sim((X_1, Y_1), (X_2, Y_2)) = Sim_{\text{extension}} * w + Sim_{\text{intension}} * (1 - w).$$

Where $w \in [0, 1]$ is a weight that is the proportion of the extension in the whole concept.

Example 19. Continues example 5. For concept 1. ($\{U_1, U_3\}$, {Internet, Spider}) $\in BL$ and concept 5. ($\{U_4, U_5, U_8\}$, {Web Page, Spider}) $\in CL$, let $w = 0.4$. The concept similarity of concept 1 and 5 can be computed as follows:

- We can know that $U_3 \to_L U_5$, $U_1 \to_L U_8$, $U_3 \to_C U_4$ and $U_3 \to_C U_5$ from Table 3. The extension similarity is 4/(4*3) =0.3333.
- The intension similarity is (1/2)*(*ics*(Internet, Webpage) + *ics*(Spider, Spider)) =0.91755.
- $Sim(1, 5) = 0.3333 * 0.4 + 0.6 * 0.91755 = 0.68385$.

For a given web page (URL $\in CL$), its concept set *Cp* includes all of the concepts in which the objects of each concept contain the URL and each concept belongs to *CL*. According to the user query, each URL of the initial URLs can stand for a user's search requirement. The concepts of *BL* are derived from the initial URLs, step by step. The degree of semantic similarity of the URL $\in CL$ with *BL* reflects the semantic relationship between the URL and the user query *Q*. This allows us to define the semantic similarity of a concept in *CL* as follows:

Definition 20. (The semantic similarity of a concept in *CL*). Consider two concept lattices *BL* and *CL*; $(X_1, Y_1) \in CL$. Let $Sim((X_1, Y_1), BL)$ be the semantic similarity of the concept (X_1, Y_1).

$$Sim((X_1, Y_1), BL) = \max_{(X,Y)\in BL} sim((X, Y), (X_1, Y_1)).$$

To make the choice of web pages for next step in web crawler efficient, each web page within the current URLs should be assigned a rank. The semantic ranking of the web pages (URLs) in concept lattice *CL* are computed as follows.

Definition 21. Consider two concept lattices *BL* and *CL*, $(X, Y) \in CL$. Each URL is an object of *X*. Let *SemRank*(URL) be the semantic rank of a URL.

$$SemRank(\text{URL}) = \max_{(X,Y)\in Cp} Sim((X, Y), BL).$$

3.5. Crawler's choice of a web page from queued URLs

A web crawler is an important tool of a search engine. The web crawler adopts a depth-first search, a width-first search and a best-first search for retrieving web pages. To choose web pages, a web crawler uses two queues (a running queue and a waiting queue)(Du et al., 2004). In contemporary search engines, the ranks of the web pages of the waiting queue are computed by the PageRank and HITS algorithms. The web pages with the highest ranks are chosen for the running queue. However, the PageRank and HITS algorithms are designed on the linkage structure of every web page. In this paper, we make use of the *SemRank* and let a web crawler base the choices for the web pages in its waiting queue on their *SemRank*. We give a threshold (γ) for determining whether a web page is relevant to a user query. If the *SemRank* is greater than γ, the URL (corresponding to the Web page) is placed into the running queue.

4. Experimental results analysis and evaluation

To evaluate our proposed semantic ranking system, a BFCrawler, CosTFCrawler, LCGCrawler, RCGCrawler and our web crawler that uses formal concept analysis were programmed under Windows XP, CPU: PM 1.7 G, memory size: 512 M. Our web crawler that uses formal concept analysis was stepped as follows:

(1) A user inputs some queries and we call Google API to gain some URLs for the initial URLs (the dead links are discard).
(2) For the downloaded web pages of the corresponding URLs, we compute their TF and design the inverse-index data structure. We acquire some important key words by sorting by the TF-IDF values.
(3) We construct the formal context (with the initial URLs as objects and the key words as attributes) and concept lattice *BL* with the algorithm proposed by Du et al. (2007).
(4) The external links of the initial URLs are parsed and their web pages are downloaded to the local computer. These URLs are pushed into the waiting queue.
(5) We construct the formal context (with the initial URLs of the waiting queue as objects and their key words as attributes) and concept lattice *CL* by using the same algorithm as used for (3).
(6) Our similarity method is used to compute the similarity of concepts and rank the URLs in the queue of URLs. The ranking of URLs (web pages) guides the direction of the web crawler and pushes them into the running queue.
(7) We let $BL \Leftarrow CL$, Loop (4) until a given the number of iterations.

Table 4
14 topics and their number of web pages for a sport data set.

	Topic	Number		Topic	Number
1	Rugby	475	8	Mixed martial arts	34
2	Football	383	9	Boxing	54
3	Baseball	662	10	Olympics	68
4	Basketball	642	11	Auto Racing	343
5	Tennis	90	12	Golf	143
6	Hockey	384	13	Rubbish page	309
7	Cycling	101	14	Others	1049

To ensure fairness within the evaluation, we gave the same initial URLs to the BFCrawler, CosTFCrawler, LCGCrawler and RCGCrawler that we gave to our web crawler.

Precision and recall are two important indices used to evaluate the performance of web crawlers. The precision is percentage of the crawled web pages that are relevant. The recall is the percentage of the relevant web pages found by a web crawler compared to all of the relevant web pages on the entire Web. Generally, these two indices are calculated as follows:

$$\text{Precision} = \frac{|R \bigcap D|}{|D|} \qquad \text{Recall} = \frac{|R \bigcap D|}{|R|}. \tag{7}$$

Where R is the set of relevant web pages that satisfy the user query, which can be obtained by *Google* and then marked by users. D is web page set, which is acquired by visiting web page resources identified by a user query. Another evaluation method is the *F*-Measure (Shaw et al., 1997), which is a traditional information retrieval performance evaluation method based on a document library. In most cases, the higher the precision is, the lower recall is, and vice versa. However, the *F*-Measure (F) considers precision and recall, comprehensively. It evaluates the retrieval performance of a retrieval system by adopting a unified measurement:

$$F = \frac{2\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

The value of F is in [0, 1] and increases as precision and recall increase. The higher F is, the better the integrated performance of a system is. With the *F*-Measure, there is no bias between precision and recall. The average harvest rate is the proportion of the number of relevant pages to the number of the downloaded pages.

In our experiments, we downloaded 5000 web pages from 14 topics of a sport dictionary in Yahoo to construct a data set (Table 4). The data set was used to measure precision and recall of a BFCrawler, CosTFCrawler, LCGCrawler, RCGCrawler, and our web crawler. To compute their precision and recall, the relevant web pages were marked by 30 volunteers in our Lab. We divided 30

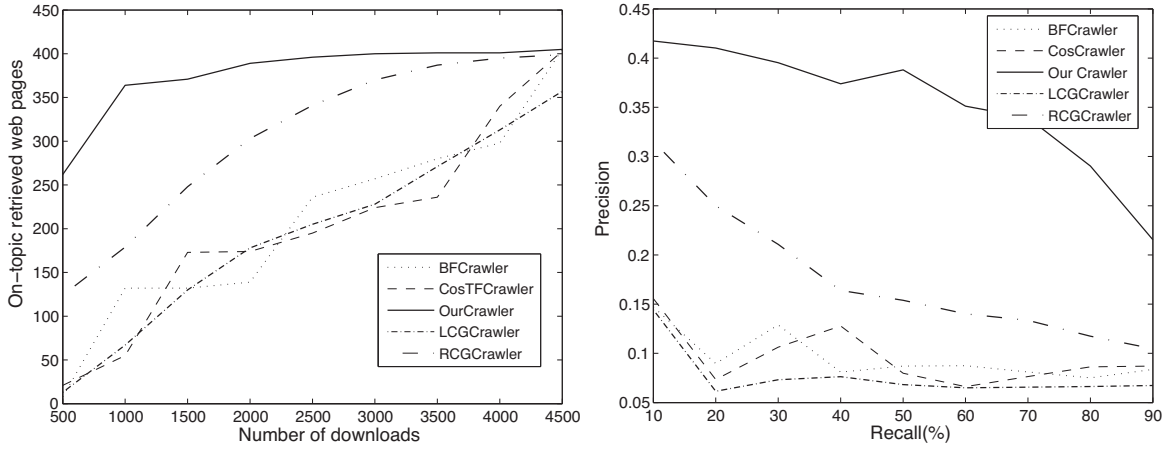

Fig. 5. The tendencies of average harvest rate, precision and recall for "Auto Racing".

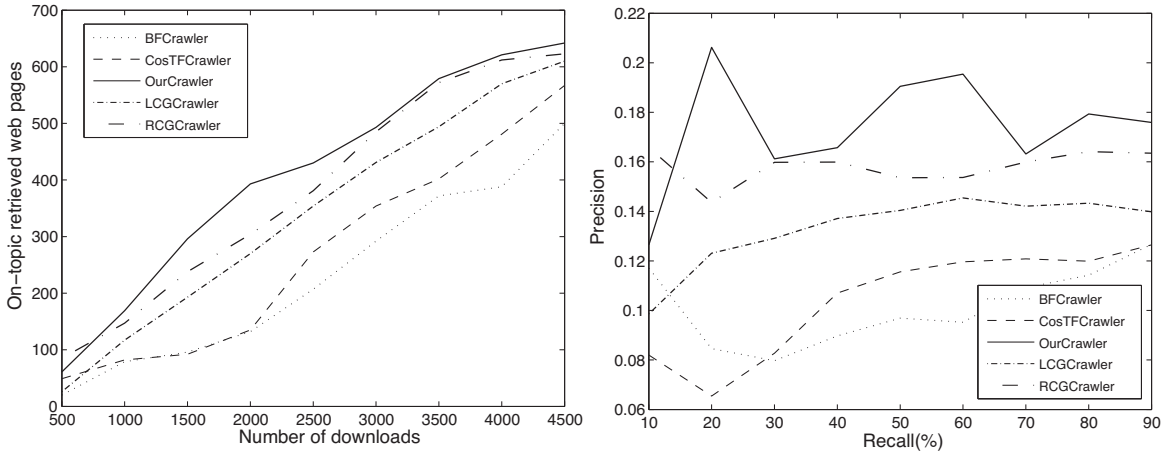

Fig. 6. The tendencies of average harvest rate and precision-recall for "Basketball".

volunteers into 10 teams. Each team was responsible for the annotation of 500 web pages in data set (Table 4), each web page and its URL was placed into a certain topic class. Our PISE have saved 557,000 web logs, it included 18,532 web logs about topic sports, we constructed the Weighted ISA and Part-Of hierarchy of topic sports. In crawling process of our web crawler, the threshold is an important parameter to distinct whether the founded web page is related or not. We designed, respectively, 10 experiments ($\gamma = 0.1, 0.2, \ldots, 1.0$) for 14 topics in Table 4. These experiments demonstrated that the outstanding performance of our web crawler is obtained when SemRank threshold γ is between 0.4 and 0.5. We set it to 0.45 in following experiments.

We chose the topic "Auto Racing" and obtained its average harvest rate (Fig. 5left) and precision–recall (Fig. 5right) to make compare our web crawler with a BFCrawler, CosTFCrawler, LCGCrawler, RCGCrawler. As shown in Fig. 5, the average harvest rate of our web crawler increases as the number of web pages crawled increases. The tendencies of the other four web crawlers are similar to our crawler. As a whole, our web crawler performs better than the others. The precision–recall relationship shown in Fig. 5 indicates that our web crawler's precision is higher than those of the BFCrawler, CosTFCrawler, LCGCrawler and RCGCrawler. This demonstrates that our web crawler is more suitable for crawling focused web pages than the other four crawlers. We also chose topic "Basketball" and obtained its average harvest rate (Fig. 6top) and precision–recall relationship (Fig. 6bottom). The results are the same as for the topic "Auto Racing".

Fig. 7 shows the average tendencies of average harvest rate and precision–recall relationship for all 14 sports topics considered. As shown in Fig. 7, the tendency of average harvest rate of our web crawler indicates that our web crawler performs better than others. The precision–recall relationships show in Fig. 7 indicate that our web crawler's precision is higher than those of the BFCrawler, CosTFCrawler, LCGCrawler and RCGCrawler. These results demonstrate that our web crawler is more suitable for crawling focused web pages than the other web crawlers. The tendencies of the F-Measure and recall are shown in Fig. 8, The F of our web crawler is higher than those of other four. The average F of the BFCrawler is 0.19, and that of the CosineTFC is 0.12. Our crawler set F to 0.25. This proves that our web crawler outperforms the BFCrawler, CosTFCrawler, LCGCrawler and RCGCrawler.

On the other hand, we chose the topic "Football" and obtained its crawling time under the same recall (Fig. 9) to make compare our web crawler with a BFCrawler, CosTFCrawler, LCGCrawler, RCGCrawler. As shown in Fig. 9, the crawling time of our web crawler increases as recall increases. The tendencies of the other four web crawlers are similar to our crawler. As a whole, the crawling time of our web crawler is less than the others under recall$\leq$0.75. When recall>0.75, Our Crawler costs too many time to construct concept lattice and compute the semantic rank of a URL, and so, the crawling time of our crawler increases faster than the other four web crawlers.

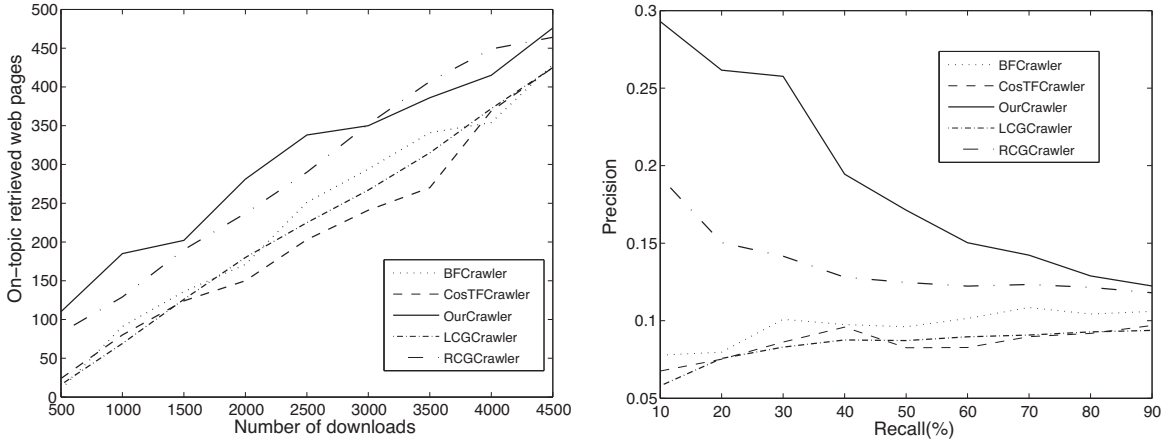

Fig. 7. The tendencies of average harvest rate and the precision–recall relationship for all topics.

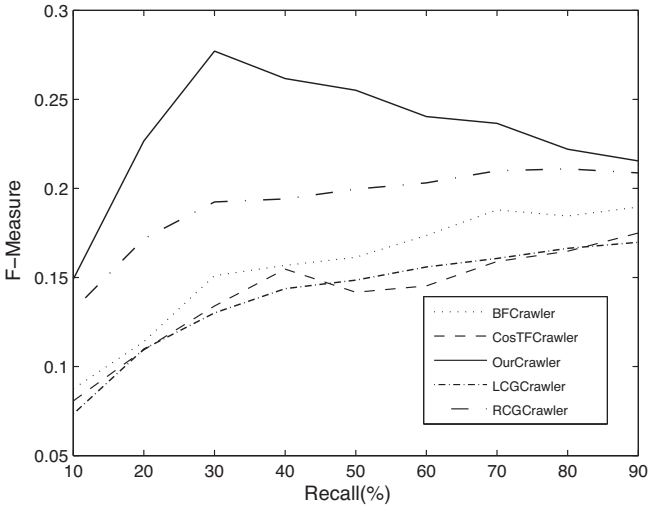

Fig. 8. The tendencies of the *F*-measure and recall for all topics.

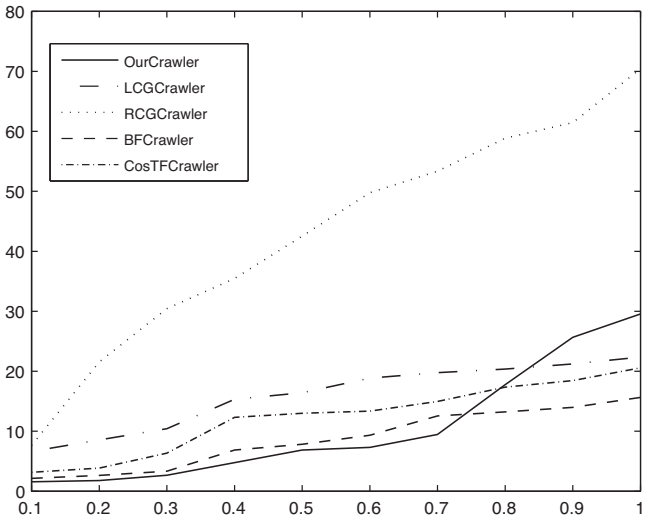

Fig. 9. The tendencies of the crawling time for "Football".

5. Conclusion

In this paper, we developed a semantic ranking system for web pages based on formal concept analysis. Methods for determining the base concept lattice and the current concept lattice of web pages are proposed. First, by analyzing a user's browsing pattern and hyperlinks, the extension similarity and intension similarity are determined. Second, by constructing an ISA and Part-Of hierarchy, the information content similarity between two nouns is computed automatically by using a user's web log. A method for computing the semantic similarity between two concepts in two different concept lattices (*BL* and *CL*) is proposed. The semantic ranks of web pages are then used. Our experimental testing demonstrates that our web crawler is more suitable for crawling focused web pages. It proved that the semantic ranks of web pages are useful and efficient for making a web crawler's choice of web pages for continuing work. In our future work, we will examine the effect of using different methods for computing intension similarity. Especially, a comparison of the different semantic ranks associated with intension similarities determined using a user's web log, Wordnet, Conceptnet and Cyc is important in our research.

Acknowledgements

This work was supported by the National Nature Science Foundation (Grant No. 60872089), the Cultivating Foundation of Science & Technology Leaders of Sichuan Province.

References

Allan, B., Roberts, G.O., 2001. Finding authorities and hubs from link structures on the World Wide Web. In: Proceedings of the 10th International World Wide Web Conference, pp. 415–429.
Almpanidis, G., Kotropoulos, C., Pitas, I., 2007. Combining text and link analysis for focused crawling. An application for vertical search engines? Information Systems 32 (6), 886–908.
Bain, M., 2003. Inductive construction of ontologies from Formal Concept Analysis. In: Proceedings of the Australian Conference on Artificial Intelligence, pp. 88–99.
Batsakis, S., Petrakis, E.G.M., Milios, E., 2009. Improving the performance of focused web crawlers. Data and Knowledge Engineering 68 (10), 1001–1013.
Brin, S., Page, L.,1998. The anatomy of a large-scale hypertextual web search engine. In: Proceedings of the 7th ACM-WWW International Conference. ACM Press, Brisbane, pp. 107–117.
Cho, J., Garcia-Molina, H., Page, L., 1998. Efficient crawling through url ordering? Computer Networks 30 (1–7), 161–172.
Cohn, D., Chang, H.,2000. Probabilistically Identifying Authoritative Documents. In: Proceedings of the Seventeenth International Conference on Machine Learning. Stanford, CA.
Conesa, J., Storey, V.C., Sugumaran, V., 2008. Improving Web-query processing through semantic knowledge. Data and Knowledge Engineering 66, 18–34.
Diligenti, M., Coetzee, F.M., Lawrence, S., et al.,2000. Focused crawling using context graphs. In: Proceedings of the 26th international coference on very large databases. VLDB, pp. 527–534.
Ding, Y., Fensel, D., Klein, M., Omelayenko, B., 2002. The semantic web: yet another hip? Data and Knowledge Engineering 41 (2-3), 205–227.
Du, Y.J., Yan, B., Song, L., 2004. Design of crawler's algorithm and Implement of crawler's program. Journal of Computer Applications 1, 33–35.
Du, Y.J., Li, H.M., Pei, Z., Peng, H., 2005. Intelligent spiders algorithm of search engine based on keyword? ECTI Transactions on Computer and Information Theory 01 (01), 40–49.
Du, Y.J., Pei, Z., Li, H.M., Xiang, D., Li, K., 2007. New Fast Algorithm for Constructing Concept Lattice. In: Proceedings of ICCSA, vol. 2, pp. 434–447.
Du, Y.J., Dong, Z.B., 2009. Focused web crawling strategy based on concept context graph. Journal of Computational Information Systems 3, 47–49.
Du, Y.J., Li, H.M., 2010. Strategy for mining association rules for web pages based on formal concept analysis? Applied Soft Computing 10 (3), 772–783.
Formica, A., 2006. Ontology-Based Concept Similarity in Formal Concept Analysis? Information Sciences 176 (18), 2624–2641.
Formica, A., 2008. Concept similarity in formal concept analysis: an information content approach? Knowledge-Based Systems 21 (1), 80–87.
Hsu, C.C., Wu, F., 2006. Topic-specific crawling on the Web with the measurements of the relevancy context graph. Information Systems 31, 232–246.
Jones, K.S., 1979. Search term relevance weighting given little relevance information. Journal of Documentation 35 (1), 30–48.
Jung, J.J., 2009. Towards open decision support systems based on semantic focused crawling. Expert Systems with Applications 36, 3914–3922.
Konstantin, A., Alexander, D., Valentina, K., Philippe, N., Olga, S., 2011. Optimal threshold control by the robots of web search engines with obsolescence of documents? Computer Networks 55 (8), 1880–1893.
Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment? Journal of ACM 46 (5), 604–632.
Lempel, R., Moran, S., 2000. The stochastic approach for link-structure analysis (SALSA) and the TKC effect? Computer Networks 33 (1), 387–401.
Liu, H.Y., Janssen, J., Milios, E., 2006. Using HMM to learn user browsing patterns for focused Web crawling. Data and Knowledge Engineering 59, 270–291.
Lin, D.,1998. An Information-Theoretic Definition of Similarity. In: Proceedings of the International Conference on Machine Learning. Morgan Kaufmann, Madison, Wisconsin, USA, pp. 296–304.
Park, J.S., Oh, S.J., Ahn, J.H., 2011. Ontology selection ranking model for knowledge reuse? Expert Systems with Applications 38 (5), 5133–5144.
Resnik, P., 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: Proceedings of the International Joint Conference on Artificial Intelligence, vol. 1, Morgan Kaufmann, Montreal, Quebec, Canada, pp. 448–453.
Rungsawang, A., Angkawattanawit, N., 2005. Learnable topic-specific Web crawler. Journal of Network and Computer Applications 28, 97–114.
Salton, G., Wong, A., Yang, C.S., 1975. A vector space model for automatic indexing. Communications of ACM 18 (11), 613–620.
Shaw, W.M., Burgin, R., Howell, P., 1997. Performance standards and evaluations in IR test collections: cluster-based retrieval models? Information Processing and Management 33 (1), 1–14.
Sufyan Beg, M.M., 2005. A subjective measure of Web search quality. Information Sciences 169, 365–381.
Uschold, H., Gruninger, M., 1996. Ontogies: principles, methods and applications. The Knowledge Engineering Review 11. (2).
Wille, R., 1982. Restructuring the lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (Ed.), Ordered Sets. Reidel, Dordrecht/Boston, pp. 445–470.

Wille, R., 1989. Lattices in data analysis: how to draw them with a computer. In: Algorithms and Order. Kluwer Acad. Publ., Dordrecht, pp. 33–58.
Wille, R., 1992. Concept lattices and conceptual knowledge systems? Comput. Math. Appl. 23 (6-9), 493–515.
Yang, S.Y., 2010. OntoCrawler: a focused crawler with ontology-supported website models for information agents? Expert Systems with Applications 37 (7), 5381–5389.
Yang, Y.K., Du, Y.J., Sun, J.Y., Hai, Y.F., 2008. A topic-specific Web crawler with concept similarity context graph based on FCA. In: Proceedings of the 4th International Conference on Intelligent Computing, LNAI, pp. 840–847.
Zheng, H.T., Kang, B.Y., Kim, H.G., 2008. An ontology-based approach to learnable focused crawling. Information Sciences 178, 4512–4522.

YaJun Du was born in February 1967. He received D.Sc. in traffic information engine and control from SWJTU (2005). Now he is a professor of XHU (Xihua University) in Computer Science. He has published several papers and served on program committees of both China and International Conferences. His experience and research work focus on information retrieve, software engineering, search engine, Web mining, computer network.

YuFeng Hai was born in January 1979. Now He is a D.Sc. student at the Computer Science Department of UESTC, China. He obtained her M.S. degree in computer science at XHU (Xihua University) in 2007. His experience and research work focus on information systems, software engineering, search engine, Web mining.