Available online at www.sciencedirect.com
ScienceDirect
Applied Soft Computing 7 (2007) 398–410
Applied Soft Computing
www.elsevier.com/locate/asoc

Personalized mining of web documents using link structures and fuzzy concept networks

Kyung-Joong Kim *, Sung-Bae Cho

Department of Computer Science, Yonsei University, 134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, Republic of Korea

Received 30 July 2004; received in revised form 1 August 2005; accepted 24 September 2005

Abstract

Personalized search engines are important tools for finding web documents for specific users, because they are able to provide the location of information on the WWW as accurately as possible, using efficient methods of data mining and knowledge discovery. The types and features of traditional search engines are various, including support for different functionality and ranking methods. New search engines that use link structures have produced improved search results which can overcome the limitations of conventional text-based search engines. Going a step further, this paper presents a system that provides users with personalized results derived from a search engine that uses link structures. The fuzzy document retrieval system (constructed from a fuzzy concept network based on the user's profile) personalizes the results yielded from link-based search engines with the preferences of the specific user. A preliminary experiment with six subjects indicates that the developed system is capable of searching not only relevant but also personalized web pages, depending on the preferences of the user.

Keywords: Web mining; Fuzzy concept networks; Personalization; Link-based search; Search engines

1. Introduction

Intelligent web will need new tools and infrastructure components in order to create an environment that serves its users wisely. Some of these components include agent technology, ontology, and mining techniques [1]. For example, researchers have already exploited structural similarities between the web and social networks in order to develop techniques that enhance web searches and trawl the cyber community [2]. To supplement keyword-based indexing, researchers have applied mining link structures to web page ranking systems [3]. Researchers in traditional computational intelligence topics can help develop intelligent, user-amenable Internet systems such as machine learning, natural language processing, and recommendations [4]. With respect to web intelligence, the construction of intelligent web search engines using computational intelligence and mining techniques is an interesting research issue.

* Corresponding author.
E-mail addresses: kjkim@cs.yonsei.ac.kr (K.-J. Kim), sbcho@cs.yonsei.ac.kr (S.-B. Cho).

1568-4946/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2005.09.002

Conventional search engines include Yahoo, Lycos and Altavista, which are some of the most important services available on the Internet [5–7]. Text-based search engines rank documents using the position and frequency of keywords for their heuristics. The more instances of a keyword and the earlier in the document those instances occur, the higher the document's ranking in the search engine results. For example, to obtain the most representative website about "physics," a text-based search engine would present a list of websites that contain the largest frequency of the word "physics." However, the content of these websites might be different from the expectations of the user. Also, web page designers can use keyword spamming to trick search engines into placing their pages at a higher rank. For example, rank spammers often stuff keywords into invisible text and tiny text. Hidden from most web users but visible to spiders, such text contains repeated instances of keywords, thereby elevating that site's rank above the rank of more scrupulous sites that restrict such keywords to legitimate usage [8]. Link-based search engines attempt to find the most authoritative sites so that these problems can be solved. There is optimism that the use of link information can help improve the quality of the results returned by these search engines [9–11].

Recently, Google and Clever Search have been considered as promising next-generation search engines [12,13]. Google and Clever Search share the common feature of using link structures. While the computation of a given web document's importance and ordering of search results are based on link structures, link information distills valuable documents that cannot be found using text information. The search results should yield the most appropriate sites in terms of the expectations of the user. Ranking web documents using link information helps search engine designers solve one of the problems that are mentioned above. Clever Search distills a large search topic on the WWW down to a size that will make sense to a human user. It identifies authoritative and hub sources about the user's query [13]. While both authoritative and hub sources are calculated using link information, authoritative sources lead to the most reliable websites about specific topics and hub sources refer to documents which link to many authoritative sources [10].

Building a personalized search engine that uses link structures is a very challenging task. Because of the size of the web, it is not easy to make consistent ranking rules for all web documents for all users. A personalized robust ranking rule that can produce proper ordering of web documents is needed. Using effective indexing methods and personalized ranking algorithms, search engines can produce appropriate results before the user's patience runs out. To develop a good personalized search engine, the designer must be aware of the above points.

This paper introduces a system that searches web documents based on link information (yielding higher quality results) and fuzzy concept networks, which yield more personalized results and more satisfaction to users [14,15]. A fuzzy concept network is able to calculate the relevance between a wide range of concepts using fuzzy logic. The network then represents the knowledge of the user [16–18]. The construction of a fuzzy concept network is based on a user profile. The search engine is able to select fitting websites for the user's query by processing fuzzy document retrieval using the fuzzy concept network to represent the user's knowledge. The fuzzy concept network and fuzzy document retrieval system are used for personalization methods, as shown in Fig. 1.

The motivation of this research is to find a way to apply a fuzzy concept network as a user profile to link-based search engines. Because the fuzzy concept network provides the inference mechanism to calculate undefined relationships between concepts, it can work based on partially specified information by the user. Undefined information can be calculated using a transitive closure of the fuzzy concept matrix. This property can minimize the user's cognitive load to insert the relevance of concepts. Although this approach suffers from the inconvenience of obtaining

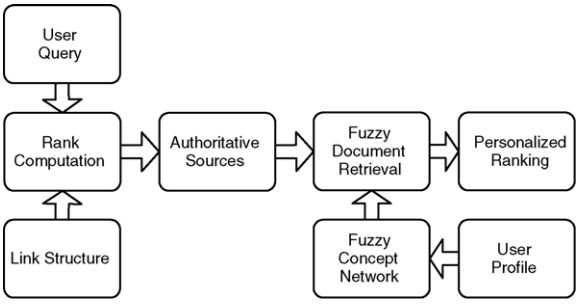


Fig. 1. Schematic diagram of personalized search engine.

information from the user, it can be improved by the use of a fuzzy concept network.

The rest of this paper is organized as follows. In Section 2, the current status of existing search engines is introduced. In Section 3, we propose the architecture of personal web search engines that use link structures and fuzzy concept networks. In Section 4, we show some preliminary experimental results and personalization processes. Conclusions and future work are discussed in Section 5.

2. Related works

Usually, a search engine consists of a crawler, an indexer, and a ranker [19]. A crawler retrieves documents from the web [20]. Search engine indexers create maps of the web by indexing web pages according to keywords. From the enormous databases that these indexes generate, search engines link the page contents through keywords to URL's (Inverted Index). When a user submits a keyword or phrase, the database of the search engine ideally returns a list of relevant URL's.

The crawler usually starts with a historical list of URL's. Such lists favor documents that contain many links, such as server lists, ''What's New'' pages, and other popular websites. The crawler retrieves documents from the web by automatically traversing its hypertext structure. It first retrieves a document and recursively retrieves all documents referenced in it. What routes these crawlers follow, which sites they visit, how often, and other operational details tend to be steadfastly guarded trade secrets. Cho et al. have explored the nature of crawling algorithms in detail [21]. Their work describes the metrics that crawlers commonly use to determine the importance of a given web page.

Unless a query term or string steers the crawler, metrics decide the importance of a given page. The Backlink metric uses a page's backlink or inlink count as an important heuristic. The backlink count refers to the number of links to a specific page that appear across the entire web. This metric derives from the assumption that the more pages that link to a given page, the greater that page's importance. The PageRank metric gives prominence to backlinks from pages that themselves have high backlink counts [22]. In theory, this approach mirrors the academic practice of giving greater weight to citations from more authoritative works. To determine the crawler's next step, the location metric uses URL location information, such as domain type and whether certain keywords such as ''home'' appear in the URL.

There are different techniques adopted by search engines to help users shift large sets of retrieved results. Nearly every search engine uses a form of ranking to present results in order of relevancy to the user. Many search engines like Excite provide a ''List by Website'' option that helps users to locate documents from a particular site. They also provide options like ''More like this'' which helps a user to identify related documents. A similar approach is used by the Northern Light search engine, which provides ''Custom Search Folders'' to organize retrieved documents [23]. These folders can be based on source (e.g., commercial websites, domains, etc.), type (e.g., press releases, product reviews, etc.), subject (e.g., hyper-tension, baseball, etc.) or language (e.g., English, French, etc.). Northern Light performs pre-clustering of its pages by assigning, at the time of indexing, a set of potential folders to each document. It then decides which folders to present at the query time [23]. A new technique is to present the results after doing dynamic document clustering of online search outputs. Grouper uses a similar post-clustering concept in its presentation of results [24]. Google and Clever Search use link structures to present their results in order of relevancy.

Google is a large-scale search engine system that makes heavy use of link structures presented in hypertext. Google is designed to crawl and index the web efficiently and produce more satisfying search results than those of existing systems. The results of a Google search are derived from full text and hyperlink databases of at least 1.3 billion pages, and can also include URL's that have not been crawled, and even broken web pages — although it does exclude some broken web pages by computing their PageRank value.

Although Clever Search does not provide commercial services, it is a promising next-generation search engine. Developed by IBM, Clever Search incorporates several algorithms that use hyperlink structures to discover high quality information on the web. This search engine also includes enhancements such as the hypertext-induced topic search (HITS) algorithm, hypertext classification, focused crawling,

mining communities, and modeling the web as a graph. A number of algorithmic methods used to improve the precision and functionality of the basic HITS algorithm have been researched in Almaden and elsewhere [25]. Using hypertext classification and topic distillation tools, a crawler can work within a specific topic domain, ignoring unrelated and irrelevant materials [26].

There are many different approaches used to personalize web searches, such as profiling and personalized ranking. Liu et al. proposed a novel technique to obtain user profiles from their search histories [31]. A user profile and a general profile can be stored from the user's search history and category hierarchy, respectively. Several profile learning and category mapping algorithms, as well as a fusion algorithm, are provided. Jeh and Widom explored how to extend the PageRank values with personalized aspects [33]. To achieve personalization, the algorithm needs specific inputs such as a set of personalized hub pages with high PageRank values to drive the computation. Nejdl and coworkers [32] automated the hub selection process of personalized PageRank algorithms using the bookmarks of the user. Singh and Dey proposed the design of a customized document-filtering scheme using rough-set theoretic approach [35]. Table 1 summarizes the related works.

3. Personal link-based search engines

Fig. 2 shows the architecture of a personal web search engine which uses hyperlink structures and fuzzy concept networks. Search engines perform crawling, storing of link structures, ranking, and personalization processes. They use only link information to find relevant web pages, so that the Store Server stores the link structure for efficient searching. The crawler extracts the link information from crawled web pages, and then sends the URL and link information to the Store Server. As the user submits a query, the search engine executes a ranking algorithm that is able to find authoritative and hub sources. The fuzzy document retrieval system is responsible for the personalization process of the authoritative sources which relate to the user's query. The fuzzy concept network can be generated for each user by using the information from the relevant user profile. Using a generated fuzzy concept network, the fuzzy document retrieval system is able to produce personalized document rankings for the user.

3.1. Crawling

To extract link information, search engines crawl web documents. Crawling starts from the initial page of a given website and recursively retrieves web documents in breadth-first order. The crawling process is as follows (Fig. 3):

1. The URL Server sends the URL that is popped from a queue in the Store Server to the crawler.
2. The crawler retrieves the web documents from the WWW.
3. The crawler analyzes the web documents and extracts the link information. The crawler sends the retrieved URL's to the queue in the Store Server.
4. Until the queue is empty, the crawler repeats steps 2 through 4.

3.2. Construction of the base set

The Store Server stores URL's and link information. A URL is represented with a unique *DocID* (Document ID) as a form <*DocID*, *URL*> and is stored in the <*DocID*, *DocID*> form. Fig. 4 shows the

Table 1
Summary of related works

Authors	Contributions	Relationship with our research
Kleinberg [10]	Propose a new link-based search algorithm	The algorithm is used to implement link-based search algorithm
Chen et al. [16,17]	Propose a fuzzy concept network as a knowledge base for document retrieval	It provides basic foundation of fuzzy concept network
Brin and Page [9]	Propose an architecture of link-based search engine with PageRank	Our link-based search engine replaces the PageRank as Kleinberg's algorithm

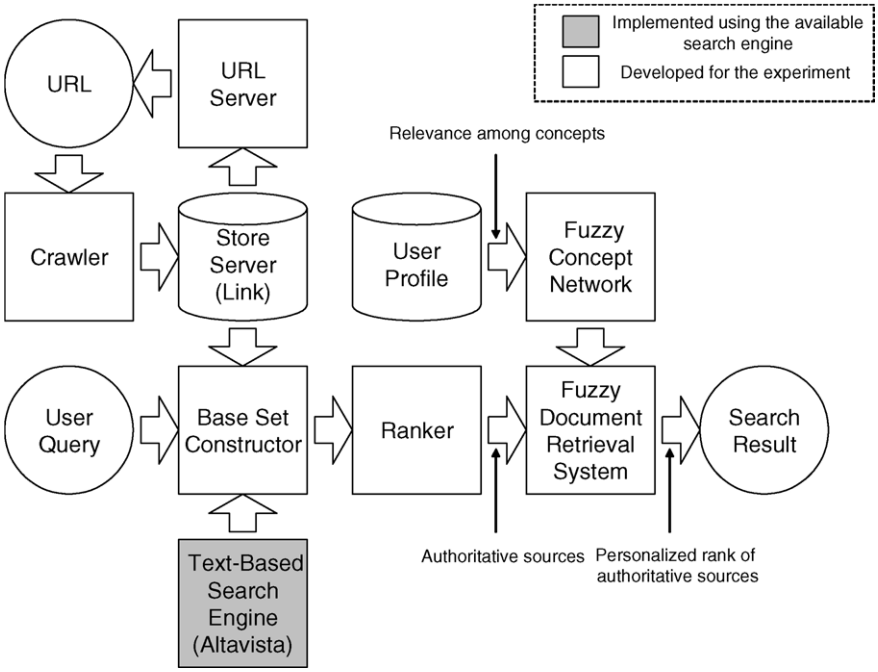


Fig. 2. Personal search engine system.

conceptual process of storing link information. To find the forward link and the backlink of URL's retrieved from text-based search engines, the search engine uses the link information stored in the Store Server. To find the forward link of the URL W_1, it gets the *DocID* of W_1 from the Store Server and then compares the *DocID* of W_1 with the starting *DocID* in the link table. If the *DocID* of W_1 is equal to the starting *DocID* in the link table, the URL of the end *DocID* becomes the forward link of W_1. Similarly, it searches the backlink of W_1 from the link table. If the end *DocID* of the link table is equal to the *DocID* of W_1, the starting *DocID* becomes the backlink of W_1

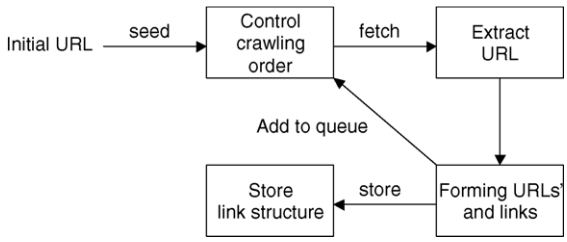


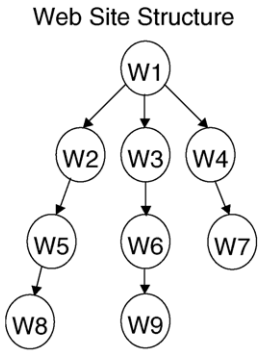
Fig. 3. The crawling process.

After finding all forward links and backlinks of the root set, the search engine constructs a base set including a root set, a forward link set, and a backlink set, as shown in Fig. 5. The ranker then calculates the authoritative and hub weights of the URL's in the base set. The link structure of the base set is needed to compute the ranking. If W_1 and W_2 are the base set documents, the Store Server returns a *DocID* for each URL. Using the link table in the Store Server, the existence of the link from W_1 to W_2 is revealed. If a link from W_1 to W_2 exists and is represented by $<i, j>$, the Store Server can find $<i, j>$ from the link table.

3.3. Link-based ranking

3.3.1. Authoritative and hub sites

It is interesting to mine the web's link structures in order to identify authoritative web pages. Table 2 shows some examples of authoritative and hub websites. Search engines can automatically identify authoritative web pages using hyperlinks. Hyperlinks contain an enormous amount of latent human annotation that can help to automatically infer the level of authority [27].



URL Table

DocID	URL
00001	W1
00002	W2
00003	W3
00004	W4
00005	W5
00006	W6
00007	W7
00008	W8
00009	W9

Link Table

DocID	DocID
00001	00002
00001	00003
00001	00004
00002	00005
00003	00006
00004	00007
00005	00008
00006	00009

Fig. 4. Procedure of storing URL's and links.

3.3.2. Algorithm

Authoritative and hub documents can be identified by using the link information for qualitative searching. While authoritative documents usually contain the most reliable contents about a specific topic, hub documents contain many links to authoritative documents. Fig. 5 shows the construction of the base set from the root set. A text-based search engine is used for constructing the root set that handles the user's query. The root set contains URL's that can be used for expanding to the base set. Including the forward link and the backlink from itself expands the root set. By iterative weight updating based on the following formula, the authoritative and hub ranks of the web documents can be determined.

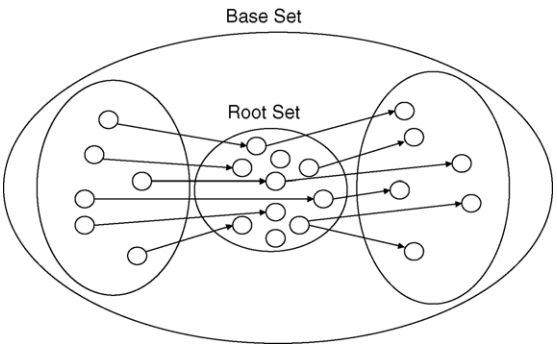


Fig. 5. Construction of the base set.

To find authoritative and hub sources in the base set, an iterative weight updating procedure is needed. The procedure is as follows:

1. If i is a document in the base set, the authoritative weight of i is a_i and the hub weight of i is h_i. a_i and h_i are initialized to 1.

Table 2
Example of authoritative and hub websites

Topic	Authoritative sites	Hub sites
Software	Homepage of Microsoft	Yahoo! Software company list
	Homepage of Oracle	Gateway to software company Web
Hardware	Homepage of Compaq	Peter's list of hardware company
	Homepage of Intel	Association of hardware company
Search engine	Homepage of Yahoo!	My favorite search engines
	Homepage of Lycos	All list of search engines in the world

2. a_i and h_i are updated by using the following formula:

$$a_i = \sum h_j(j \text{ points to } i)$$

$$h_i = \sum a_j(j \text{ pointed to by } i)$$

3. Normalize the weight of authoritative and hub websites so that the sum of the squares is 1.
4. Until authoritative and hub weights do not change, repeat steps 2 and 3.

From the converged authoritative and hub weights, the most authoritative and hub sources can be determined [10].

The root set from a text-based search engine does not contain all authoritative and hub sources relating to the user's query. By expanding the root set, the base set might contain authoritative and hub sources that are not included in the root set. The base set usually contains sufficient authoritative and hub sources relating to the user's query.

3.4. Personalization

Lucarella proposed a fuzzy concept network for information retrieval [28]. A fuzzy concept network includes nodes and directed links. Each node represents a concept or a document. $C = \{C_1, C_2, \ldots, C_n\}$ represents a set of concepts. If $C_i \xrightarrow{\mu} C_j$, this indicates that the degree of relevance from concept C_i to C_j is μ. If $C_i \xrightarrow{\mu} d_j$ this indicates that the degree of relevance of document d_j with respect to concept C_i is μ. $C_i \xrightarrow{\mu} C_j$ is represented with $f(C_i, C_j) = \mu$. Using fuzzy logic, if $f(C_i, C_j) = \alpha$ and $f(C_j, C_k) = \beta$ then $f(C_i, C_k) = \min(\alpha, \beta)$. $Ci \xrightarrow{\mu} d_j$ is represented by $g(C_i, d_j) = \mu$. A document d_j has a different relevance to concepts. A document d_j can be expressed as a fuzzy subset of concepts:

$$d_j = \{(C_i, g(C_i, d_j)) | C_i \in C\}$$

If there are many routes from C_i to C_j, $f(C_i, C_j)$ is assigned with the maximum value. Fig. 6 shows an example of a fuzzy concept network with twelve nodes. In this figure, $f(C_3, C_2) = \max(0.4, 0.3, 0.2)$ and finally becomes 0.4.

Using the fuzzy concept network, the document descriptor relating to the $d_1, d_2, \ldots, d_n$ documents can be defined. The fuzzy document retrieval system can decide the importance of the documents that use the fuzzy concept network. If a user query is equal to concept C_i, it chooses the most relevant documents relating to concept C_i among $d_1, d_2, \ldots, d_n$. Because it takes a long time to produce search results with this method, a fuzzy concept matrix with fuzzy document retrieval is used.

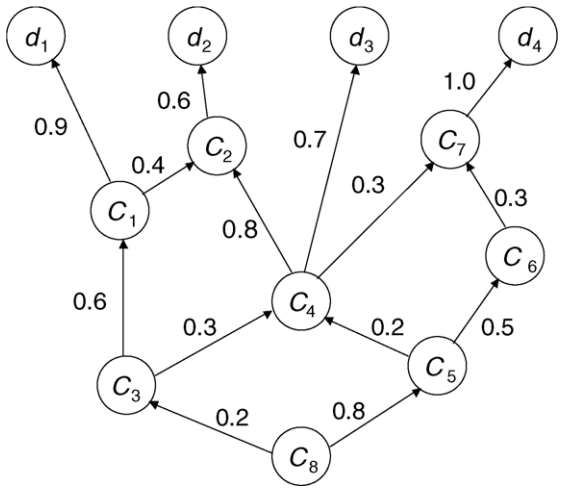


Fig. 6. A fuzzy concept network.

Meanwhile, the fuzzy document retrieval system uses fuzzy logic to deal with the uncertainty of document retrieval. Zadeh proposed the fuzzy theory in 1965 [29]. The fuzzy set theory provides a sound mathematical framework to deal with uncertainties [30]. The fuzzy document retrieval system can be defined as follows [18]:

$$< H, C, Q, I, K, \phi, \psi >$$

H is the set of documents, C, the set of concepts, Q, the set of queries, I, the binary fuzzy indexing relation from H to C, K, the knowledge base, ϕ, $Q \times H \rightarrow [0, 1]$, retrieval function, and ψ, $H \times H \rightarrow [0, 1]$, relevance function.

For each pair (q, h), $q \in Q, h \in H, \phi(q, h) \in [0, 1]$ is called the retrieval status value. For each pair (h_1, h_2), $h_1, h_2 \in H$, $\psi(h_1, h_2) \in [0, 1]$ is called the degree of relevance between h_1 and h_2 or relevance degree between h_1 and h_2. The binary fuzzy indexing relation I is represented in the form of

$$I = \{\mu_1(h, c), (h, c) | h \in H, c \in C\}$$

with a membership function μ_1: $H \times C \rightarrow [0, 1]$, indicating for each pair (h, c) to what degree the

concept c is relevant to document h. For each document $h \in H$, on the basis of the binary indexing relation I, the document descriptor I_h of h is a fuzzy subset of C defined as follows:

$$D = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{bmatrix}$$

$$d_{ij} = I_{h_i}(C_j), 1 \le i \le m, 1 \le j \le n$$

$C = \{c_1, c_2, \ldots, c_n\}$ is a set of concepts. A fuzzy concept matrix K is a matrix in which $K_{ij} \in [0, 1]$. The (i, j) element of K represents the degree of relevance from concept c_i to concept c_j. $K^2 = K \otimes K$ is the multiplication of the concept matrix:

$$K_{ij}^2 = \bigvee_{l=1}^{n} (K_{il} \wedge K_{lj}), 1 \le i, j \le n$$

$\vee$ and $\wedge$ represent the maximum and the minimum operations, respectively. Then, there exists an integer $\rho \le n - 1$, such that $K^\rho = K^{\rho+1} = K^{\rho+2} = \ldots$. Let $K^* = K^\rho$. K^* is called the transitive closure of the concept matrix K. Missing information from the fuzzy concept network can be inferred from the transitive closure of itself. The relevance degree of each document, with respect to a specific concept, can be improved by computing the multiplication of the document descriptor matrix D and the transitive closure of the concept matrix K as follows [14]:

$$D^* = D \otimes K^*$$

D^* is called the expanded document descriptor matrix.

The fuzzy document retrieval system personalizes the results obtained by link-based search engines. It selects the five most authoritative and reliable sources for a given user query. First, it defines a document descriptor using the frequency of the concept in the document. For each document, it counts the occurrence of the concepts in the user profile and normalizes the count between 0 and 1.

d_{ij} is determined automatically using the following formula. O_{hi} (C_j) is the occurrence number of C_j in document h_i:

$$d_{ij} = \frac{O_{h_i}(C_j)}{O_{h_i}(C_1) + O_{h_i}(C_2) + O_{h_i}(C_3) + \cdots + O_{h_i}(C_n)}$$

User Profile

Java	Book	0.7
Java	Car	0.3
Java	WWW	0.9
Java	Ship	0.1
Book	Car	0.3
Book	WWW	0.5
Book	Ship	0.1
Book	Cafe	0.4
Car	WWW	0.7
Car	Ship	0.6
WWW	Ship	0.5
Ship	Cafe	0.3

Fuzzy Concept Matrix

	Java	Book	Car	WWW	Ship	Cafe
Java	1.0	0.7	0.3	0.9	0.1	0.0
Book	0.7	1.0	0.3	0.5	0.1	0.4
Car	0.3	0.3	1.0	0.7	0.6	0.0
WWW	0.9	0.5	0.7	1.0	0.5	0.0
Ship	0.1	0.1	0.6	0.5	1.0	0.3
Cafe	0.0	0.4	0.0	0.0	0.3	1.0

Fig. 7. Construction of a fuzzy concept matrix based on a user profile.

The fuzzy concept matrix can be constructed from the user profile that shows some relevance between n concepts. Fig. 7 shows the construction of a fuzzy concept matrix based on a given user profile. Each user provides the system with an initial value for the fuzzy concept matrix. These values represent the user's interest with regard to the concepts. If the relevance between C_i and C_j is recorded in the user profile as μ, the $<i, j>$ element of the fuzzy concept matrix is assigned as μ. If the relevance between C_i and C_j is not recorded in the user profile, the $<i, j>$ element of the fuzzy concept matrix is assigned as 0. Transitive closure of the fuzzy concept network represents all degrees of relevances among n concepts.

The expanded document descriptor of the five most authoritative sources can be determined by multiplying the document descriptors of these documents and the transitive closure of the user's fuzzy concept network. Using the expanded document descriptor, a new ranking of the documents is generated. The sum of relevance of a document with respect to the concepts is used for the reordering of authoritative sources. Personalized ranking is based on the sum of d_{ij}^*:

$$R = \begin{bmatrix} d_{11}^* + d_{12}^* + d_{13}^* + \cdots + d_{1n} \\ d_{21}^* + d_{22}^* + d_{23}^* + \cdots + d_{2n} \\ \vdots \\ d_{m1}^* + d_{m2}^* + d_{m3}^* + \cdots + d_{mn} \end{bmatrix}$$

Kernel matrix is a similarity matrix and each entry represents a measure of similarity between two objects [34]. However, each entry of fuzzy concept network represents a measure of relevance between two con-

cepts. Though the kernel matrix is different from the fuzzy concept network, it can be useful to investigate the relationship between them.

4. Experimental results

In the experiment, the proposed search engine obtained 100 URL's from the text-based search engine, Altavista. The root set consisted of these 100 URL's. The Store Server returned the forward links and the backlinks of the root set documents. The link information in the Store Server was crawled from the web. Due to time constraints, the crawling process was restricted to a specific topic. The starting URL of the crawling process was determined as a representative website for the topic by the system designer. The base set consisted of the root set, the forward link set and the backlink set. Among the documents of the base set, the ranking algorithm found the authoritative and hub sources. To regulate the size of the base set, it limited the forward links and backlinks of the root set document to 3 and 50, respectively. It selected the first three URL's in a document as the forward links. The size of the base set was between 500 URL's and 1000 URL's. An empirical observation indicated that the authoritative and hub weights of the documents converged in less than five iterations. Therefore, the iteration number of the ranking algorithm was determined as 5. Table 3 shows the search results of the query "Java".

The search engine selected "www.java.sun.com" as the most authoritative site about "Java" Also, it selected other famous Java sites such as "www.javalobby.org," "www.javaboutique.internet.com," "www.java.about.com/compute/java/mbody.htm," and "www.javaworld.com" as authoritative sites. Table 4 shows the experimental results for other queries related to "Java." It selected "www.jini.org" as the most authoritative site about "Jini."

The personalized search engine selected the five most authoritative results as a source of personalization and produced a document descriptor of these documents. The ranking of these five documents was reordered with respect to the user's interest recorded on a user profile, which contained the following 10 concepts: "Book," "Computer," "Java," "Internet," "Corba," "Network," "Software," "Unix," "Family," and "Newspaper." The user profile contained 20 degrees of relevance between 10 concepts. A fuzzy concept network for a user was generated based on 20 degrees of relevance in the user profile, and unrecorded information was inferred from the transitive closure of the fuzzy concept network. The expanded document descriptor resulted from the multiplication of the document descriptor and the user's fuzzy concept network. The sum of the degree of relevances with respect to the concepts determined the new ranking of the documents as the final result.

In this experiment, six users evaluated five authoritative documents about "Java." Table 5 shows the rankings that the six users made. Table 6 shows the personalized results from the search engine about "Java" for the six users. Table 7 shows the rankings of the websites from Google, Altavista and AlltheWeb. We computed the average difference between the rankings performed by the users and the search engine. The average difference between the rankings was d. This is defined as follows:

$$d = \frac{1}{m}\sum_{i=1}^{m}\left|r_i - r'\right|$$

where m is the number of web pages, r_i, the ranking of the user, and r_i' is the ranking produced by the search engine. Table 8 shows the average difference between each user. The proposed method produces better performance than that produced by the general search engines.

Table 3
Search results of "Java"

Rank	Authoritative result	Hub result
1	www.java.sun.com	www.industry.java.sun.com/products
2	www.javalobby.org	www.java.sun.com/industry
3	www.javaboutique.internet.com	www.java.sun.com/casestudies
4	www.java.about.com/compute/java/mbody.htm	www.industry.java.sun.com/javanews/developer
5	www.javaworld.com	www.industry.java.sun.com/jug

Table 4
Search results of java-related queries

Query	Rank	Authoritative results
"Java2"	1	www.java.sun.com
	2	www.appserver-zone.com
	3	www.sun.com/service/sunps/jdc/java2.html
	4	www.jdc.sun.co.jp
	5	www.java.sun.com/products/jdk/1.2
"Javaone"	1	www.java.sun.com
	2	www.togethersoft.com
	3	www.javacats.com
	4	www.zdevents.com
	5	www.washington.edu/bibsys/mattf/javaone
"Jdk"	1	www.java.sun.com
	2	www.developer.netscape.com/software/jdk/download.html
	3	www.java.sun.com/products/jdk/1.1/docs/index.html
	4	www.ora.com/info/java
	5	www.kbs.cs.tu-berlin.de/~jutta/ht/JDK-beta2-quickref.html
"Jguru"	1	www.java.sun.com
	2	www.magelang.com
	3	www.javaworld.com
	4	www.java.sun.com/products/javamail/index.html
	5	www.developer.java.sun.com
"Jini"	1	www.jini.org
	2	www.java.sun.com
	3	www.artima.com
	4	www.archives.java.sun.com/archives/jinni-users.html
	5	www.sun.com/jini/news/artcliprev.html
"Servlet"	1	www.java.sun.com
	2	www.servletcentral.com
	3	www.java.sun.com/products/servlet/index.html
	4	www.archives.java.sun.com/archives/servlet-interest.html
	5	www.webmacro.org

For example, User 1 wanted important sources about "Java." The proposed search engine provided five documents about "Java" which were ordered by the weights of the authoritative sources. They were labeled as $h1$–$h5$. $C = \{C_1, C_2, \ldots, C_{10}\}$ represents a set of 10 concepts in the user profile. They are "Book," "Computer," "Java," "Internet," "Corba," "Network," "Software," "Unix," "Family," and "Newspaper."

The matrix K shows the fuzzy concept matrix of User 1. Because the user profile contained 20 degrees of relevance between 10 concepts, the unrecorded value was assigned as 0. The matrix K^* shows a transitive closure of the matrix K. Using this matrix,

Table 5
Ranking of six users (each user evaluated five documents)

	User1	User 2	User 3	User 4	User 5	User 6
www.java.sun.com	1	1	1	2	1	3
www.javalobby.org	2	2	2	1	2	5
www.javaboutique.internet.com	4	4	5	4	3	2
www.java.about.com/compute/java/mbody.htm	3	5	3	3	4	1
www.javaworld.com	5	3	4	5	5	4

Table 6
Personalized search results (shading means that the personalized rank is equal to the user's evaluation)

	User 1	User 2	User 3	User 4	User 5	User 6
www.java.sun.com	2	1	2	1	2	4
www.javalobby.org	1	2	1	3	1	1
www.javaboutique.internet.com	3	3	3	2	3	3
www.java.about.com/compute/java/mbody.htm	4	4	5	5	4	2
www.javaworld.com	5	5	4	4	5	5

Table 7
Search results yielded by Google, Altavista, and AlltheWeb (R1: ranking of the web pages in the search engine results, R2: relative ranking of the web pages among the five authoritative sources)

	Google		Altavista		AlltheWeb	
	R1	R2	R1	R2	R1	R2
www.java.sun.com	1	1	1	1	1	1
www.javalobby.org	12	4	16	4	14	4
www.javaboutique.internet.com	4	2	6	2	6	2
www.java.about.com/compute/java/mbody.htm	50	5	36	5	33	5
www.javaworld.com	5	3	8	3	8	3

Table 8
Average difference between the rankings of the user and the search engine

	User 1	User 2	User 3	User 4	User 5	User 6	Average
The proposed method	4	4	6	8	2	8	5.33 ± 2.422
Google, Altavista, AlltheWeb	8	4	8	10	6	8	7.33 ± 2.065

the unrecorded values were determined, and thereby all degrees of relevance between concepts were also determined. The document descriptor of the five authoritative "Jave" sources was D. The expanded document descriptor was calculated by multiplying a document descriptor and the fuzzy concept matrix. The expanded document descriptor was D^*. The relevance of each document with respect to the user's interest was R. The five authoritative sites were reordered as $h2$, $h1$, $h3$, $h4$, and $h5$, so that the proposed search engine provided the correct ranking for $h5$:

$$K = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \\ C_{10} \end{array} \begin{bmatrix} 1.0 & 0.9 & 0.3 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.9 & 1.0 & 0.0 & 0.2 & 0.2 & 0.9 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.0 & 1.0 & 0.5 & 0.5 & 0.8 & 0.3 & 0.9 & 0.0 & 0.0 \\ 0.0 & 0.2 & 0.5 & 1.0 & 0.2 & 0.0 & 0.0 & 0.0 & 0.7 & 0.0 \\ 0.0 & 0.2 & 0.5 & 0.2 & 1.0 & 0.4 & 0.3 & 0.8 & 0.0 & 0.0 \\ 0.0 & 0.9 & 0.8 & 0.0 & 0.4 & 1.0 & 0.0 & 0.5 & 0.6 & 0.0 \\ 0.0 & 0.0 & 0.3 & 0.0 & 0.3 & 0.0 & 1.0 & 0.1 & 0.2 & 0.1 \\ 0.0 & 0.0 & 0.9 & 0.0 & 0.8 & 0.5 & 0.1 & 1.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.7 & 0.0 & 0.6 & 0.2 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.1 & 0.0 & 0.0 & 1.0 \end{bmatrix}$$

The following matrix shows a transitive closure of the above matrix. Using this matrix, the unrecorded values were determined. All degrees of relevance between concepts were determined as follows:

$$K^* = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ C_7 \\ C_8 \\ C_9 \\ C_{10} \end{array} \begin{bmatrix} 1.0 & 0.9 & 0.8 & 0.6 & 0.8 & 0.9 & 0.3 & 0.8 & 0.6 & 0.1 \\ 0.9 & 1.0 & 0.8 & 0.6 & 0.8 & 0.9 & 0.3 & 0.8 & 0.6 & 0.1 \\ 0.8 & 0.8 & 1.0 & 0.6 & 0.8 & 0.8 & 0.3 & 0.9 & 0.6 & 0.1 \\ 0.6 & 0.6 & 0.6 & 1.0 & 0.6 & 0.6 & 0.3 & 0.6 & 0.7 & 0.1 \\ 0.8 & 0.8 & 0.8 & 0.6 & 1.0 & 0.8 & 0.3 & 0.8 & 0.6 & 0.1 \\ 0.9 & 0.9 & 0.8 & 0.6 & 0.8 & 1.0 & 0.3 & 0.8 & 0.6 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 0.3 & 1.0 & 0.3 & 0.3 & 0.1 \\ 0.8 & 0.8 & 0.9 & 0.6 & 0.8 & 0.8 & 0.3 & 1.0 & 0.6 & 0.1 \\ 0.6 & 0.6 & 0.6 & 0.7 & 0.6 & 0.6 & 0.3 & 0.6 & 1.0 & 0.1 \\ 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 1.0 \end{bmatrix}$$

The document descriptor of the five authoritative ''Java'' sources is as follows:

$$D = \begin{array}{c} \\ h1 \\ h2 \\ h3 \\ h4 \\ h5 \end{array} \begin{bmatrix} 0.0 & 0.0 & 0.2 & 0.0 & 0.0 & 0.4 & 0.4 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.0 & 0.5 & 0.2 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.2 & 0.5 & 0.0 & 0.1 & 0.1 & 0.0 & 0.1 & 0.0 \\ 0.4 & 0.2 & 0.0 & 0.1 & 0.0 & 0.1 & 0.1 & 0.0 & 0.0 & 0.0 \\ 0.3 & 0.1 & 0.2 & 0.0 & 0.0 & 0.2 & 0.2 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

The expanded document descriptor was calculated by multiplying a document descriptor and the fuzzy concept matrix. The expanded document descriptor is as follows:

$$D^* = \begin{array}{c} \\ h1 \\ h2 \\ h3 \\ h4 \\ h5 \end{array} \begin{bmatrix} 0.2 & 0.4 & 0.4 & 0.2 & 0.4 & 0.4 & 0.4 & 0.4 & 0.4 & 0.1 \\ 0.3 & 0.3 & 0.5 & 0.5 & 0.5 & 0.5 & 0.3 & 0.5 & 0.2 & 0.0 \\ 0.2 & 0.2 & 0.5 & 0.5 & 0.2 & 0.2 & 0.2 & 0.2 & 0.5 & 0.1 \\ 0.4 & 0.4 & 0.3 & 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.3 & 0.3 & 0.3 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0.1 \end{bmatrix}$$

The relevance of each document with respect to the user's interest is as follows:

$$R = \begin{array}{c} \\ h1 \\ h2 \\ h3 \\ h4 \\ h5 \end{array} \begin{bmatrix} 3.3 \\ 3.6 \\ 2.8 \\ 2.2 \\ 2.2 \end{bmatrix}$$

The five authoritative documents were reordered as $h2$, $h1$, $h3$, $h4$ and $h5$, so that the proposed search engine provided the correct ranking of $h5$.

5. Conclusions and future work

To find relevant web documents for a given user, the proposed search engine uses link structures and a fuzzy concept network. The search engine finds authoritative and hub sources for a user query using link structures. For efficient searching, these link structures are stored in advance. The fuzzy document retrieval system personalizes the link-based search results with respect to the user's interests. The user's knowledge is represented using the fuzzy concept network. The search engine finds relevant documents in which user is interested and reorders with respect to the user's interests. Future work will proceed as follows. Using the user's feedback about the search results, it is possible to change the value of the fuzzy concept network. This adaptation procedure helps to obtain better results. These preliminary results indicate that a soft computing method such as fuzzy logic can play a crucial role in information retrieval from the web,

which provides an important platform for personalization of search engines.

Acknowledgement

This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

References

[1] N. Zhong, J. Liu, Y. Yao, In search of the wisdom Web, IEEE Comput. 35 (11) (2002) 27–31.
[2] R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, The Web and social networks, IEEE Comput. 35 (11) (2002) 32–36.
[3] J. Han, K.C.-C. Chang, Data mining for web intelligence, IEEE Comput. 35 (11) (2002) 64–70.
[4] N. Cercone, L. Hou, V. Keselj, A. An, K. Naruedomkul, X. Hu, From computational intelligence to web intelligence, IEEE Comput. 35 (11) (2002) 72–76.
[5] Yahoo, http://www.yahoo.com.
[6] Lycos, http://www.lycos.com.
[7] Altavista, http://www.altavista.com.
[8] L. Introna, H. Nissenbaum, Defining the web: the politics of search engines, IEEE Comput. 33 (1) (2000) 54–62.
[9] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, in: Proceedings of the Seventh International WWW Conference, 1998. http://www7.scu.edu.au/programme/fullpapers/1921/com1921.htm.
[10] J. Kleinberg, Authoritative sources in a hyperlinked environment, IBM Research Report RJ 10076, 1997.
[11] M.R. Henzinger, Hyperlink analysis for the web, IEEE Internet Comput. 5 (1) (2001) 45–50.
[12] Google, http://www.google.com.
[13] The Clever Search, http://www.almaden.ibm.com/cs/k53/clever.html.
[14] K.-J. Kim, S.-B. Cho, Conceptual information extraction with link-based search, in: Web Intelligence 2001, Maebashi, Japan, 2001, pp. 367–372.
[15] K.-J. Kim, S.-B. Cho, A personalized web search engine using fuzzy concept network with link structure, in: Proceedings of the Joint 9th IFSA World Congress and 20th NAFIPS International Conference, vol. 1, Vancouver, Canada, 2001), pp. 81–86.
[16] S.-M. Chen, Y.-J. Horng, Fuzzy query processing for document retrieval based on extended fuzzy concept networks, IEEE Trans. Syst. Man Cybern. 29 (1) (1999) 96–104.
[17] S.-M. Chen, J.-Y. Wang, Document retrieval using knowledge-based fuzzy information retrieval techniques, IEEE Trans. Syst. Man Cybern. 25 (5) (1995) 793–803.
[18] C.-S. Chang, A.L.P. Chen, Supporting conceptual and neighborhood queries on the world wide web, IEEE Transa. Man Cybern. 28 (2) (1998) 300–308.
[19] A. Arasu, J. Cho, H. Garcia-Molina, A. Paepcke, S. Raghavan, Searching the web, ACM Trans. Internet Technol. 1 (1) (2002) 2–43.
[20] B. Pinkerton, Finding what people want: experiences with the webcrawler, in: Proceedings of the Second International WWW Conference, Chicago, USA, 1994. http://www.think-pink.com/bp/WebCrawler/WWW94.html.
[21] J. Cho, H. Garcia-Molina, L. Page, Efficient crawling through URL ordering, Comput. Networks ISDN Syst. 30 (1–7) (1998) 161–172.
[22] L. Page, PageRank: bringing order to the web, Stanford Digital Libraries Working Paper 1997-0072, 1997.
[23] Northern Light, http://www.northernlight.com.
[24] O. Zamir, O. Etzioni, Grouper: a dynamic clustering interface to web search results, Comput. Networks 31 (11–16) (1999) 1361–1374.
[25] S. Chakrabarti, B.E. Dom, D. Gibson, R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, Spectral filtering for resource discovery, in: Proceedings of the SIGIR 1998 Workshop on Hypertext IR for the Web, Melbourne, Australia, 1998.
[26] S. Chakrabarti, M. Van den Berg, B. Dom, Focused crawling: a new approach to topic specific resource discovery, in: Proceedings of the Eighth World Wide Web conference, Toronto, Canada, 1999.
[27] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2001.
[28] D. Lucarella, R. Morara, FIRST: fuzzy information retrieval system, J. Inform. Sci. 17 (2) (1991) 81–91.
[29] L.A. Zadeh, Fuzzy sets, Inform. Control 8 (1965) 338–353.
[30] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets Syst. 1 (1) (1978) 3–28.
[31] F. Liu, C. Yu, W. Meng, Personalized web search for improving retrieval effectiveness, IEEE Trans. Knowl. Data Eng. 16 (1) (2004) 28–40.
[32] P.-A. Chirita, D. Olmedila, W. Nejdl, PROS: a personalized ranking platform for web search, in: Proceedings of the 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2004), 2004, pp. 34–43.
[33] G. Jeh, J. Widom, Scaling personalized web search, in: Proceedings of the 12th International World Wide Web Conference, 2003, pp. 271–279.
[34] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, M.I. Jordan, Learning the kernel matrix with semidefinite programming, J. Mach. Learn. Res. 5 (2004) 27–72.
[35] S. Singh, L. Dey, A new customized document categorization scheme using rough membership, Appl. Soft Comput. 5 (4) (2005) 373–390.