A Novel Interdisciplinary Approach to Deep Web Search Engine

Pyke Tin, Thi Thi Zin and Hiromitsu Hama

Graduate School of Engineering, Osaka City University, Osaka, 558-8585 Japan

Summary
In this paper, we will discuss on the strategic issues of exploring, harvesting and integrating the Deep Web. We will then develop a novel interdisciplinary stochastic model for a Deep Web Search Engine which can detect and rank the contents optimally. Our efforts aim at opening up to users by building Generator. On this information grand voyage, the Generator will address the challenges of exploring, harvesting and integrating of the Deep Web. First, to make the Web systematically accessible: our Generator will focus on the discovery, modeling and structuring of databases on the Web to develop a search engine, in order to help users find sources useful for their information need. Second, to make the Web uniformly usable: the Generator will help users to make optimal choice of keywords. Based on these insights, we design a stochastic model and employ an interdisciplinary approach consisting stochastic and optimization techniques. In addition, we consider three types of keywords: text-based, image-based and hybrid-based. Experimental and simulation results are given for illustrations.

Key words:
Deep Web, Generator, search engine, stochastic model, interdisciplinary approach.

1. Introduction

We have witnessed the rapid growth of the World Wide Web. The Web has not only broadened but also deepened [1, 2]. Thus, it may be visualized that the Web has two layers: the surface layer or the Surface Web and a vastly complex layer or the Deep Web. Surface Web consists of abundant pages that can be browsed, while Deep Web contains much more data resource [3, 4]. The Deep Web pages are stored in searchable databases and these pages are created dynamically by the background databases after users submit requests to there. Conventional Search Engines cannot see or retrieve contents in the Deep Web since those pages do not exist until they are created dynamically as the result of a specific search. Because conventional Search Engine crawlers can not probe beneath the surface the Deep Web has heretofore hidden.

However, the information quantity of Deep Web is 500 times as that of surface [5], and is still increasing. Given a specific domain, such as book, movie, sports and some others, if users are interested in something, they have to submit requests or queries to all databases in the relative domain. It is time consuming and is of low efficiency. So, it is very useful to construct a unified Generator which needs to integrate databases in a domain. The Generator will become an entrance of Deep Web databases. Thus, submitting requests or queries to the Generator is the main approach to obtain information. These Generators are key components in developing modern Deep Web Search Engine. In Fig.1, we represent in a non-specific way, the improved results that can be obtained from the proposed approach. By first identifying where the proper searchable information resides, a directed keyword can then be placed to each of these sources simultaneously to harvest only the results desired with reasonable accuracy.

Bergman has described the limitation of conventional Web Search Engines which search the Web on surface but cannot search the Deep Web [2]. The approach of using graph theory for searching the Web is not new. Many approaches have been discussed how to apply such methods for the Surface Web [6, 7]. Several issues which are related to crawling the hidden Web or the Deep Web have been discussed in [8]. These issues include keyword selection, stopping conditions, determining the number of results on the Web resource, detecting error pages, detecting harvesting characteristics.

Research has been going on the hidden Web in various directions. One issue is how to find the parameters of the Deep Websites while other focuses on how to classify the Websites to their domain. Since the Deep Web resource is generated on the basis of parameters, so understanding the parameters details is very important for sending the harvesting requests and getting the response. Another important issue is how to collect or differentiate the important keywords from unimportant ones in the results returned. The results returned in response of a query will contain an entire Web page which may have important areas and unimportant sections. In this paper we consider that issue. We assume that our program will somehow extract the required information which would be meaningful keywords from the Web pages returned in the result. Related work in this area has been given by [9, 10].

Let us take the reference of a Website called papersinvited.com whose database is continuously updated by information on international conferences being held on

Manuscript received November 5, 2008
Manuscript revised November 20, 2008

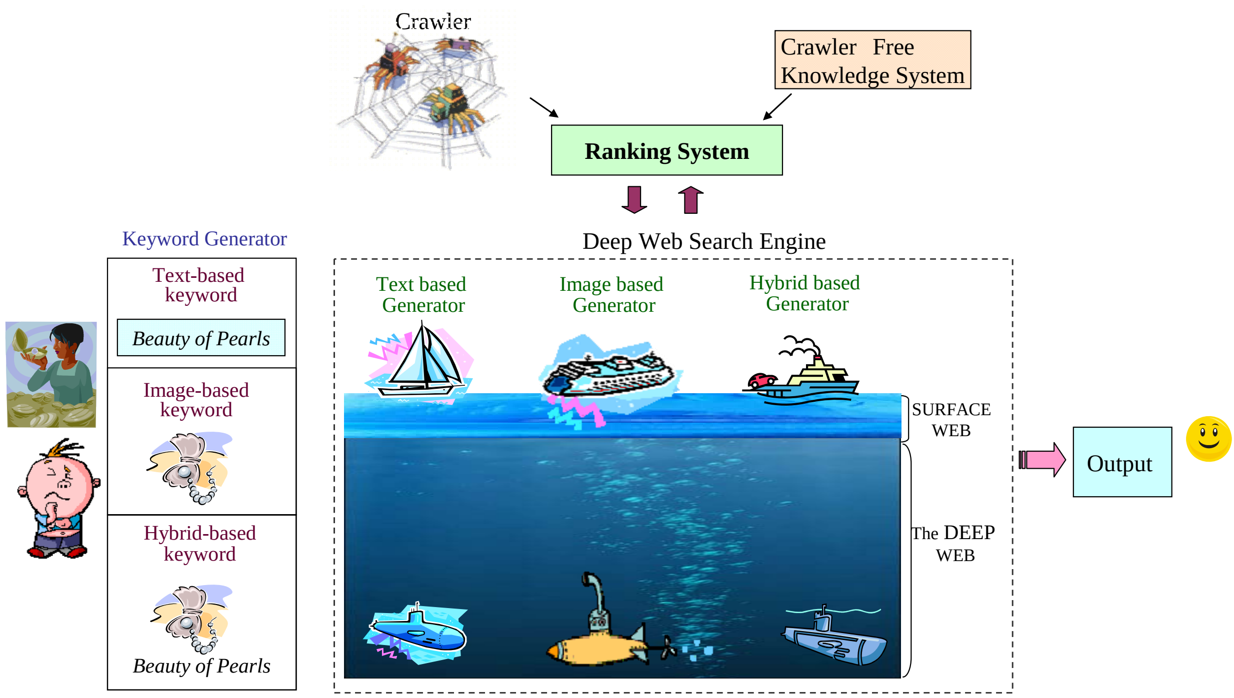


Fig. 1 Conceptual Deep Web Search Engine.

various subjects through the world. When one visits the site and searches for say a keyword 'statistics', he is presented with a list of upcoming conferences. When the user chose one, he is directed to a page that has an URL like "http://www.papersinvited.com...txkeyword=statistics". This is basically the link to the HTML page generated dynamically in response to our query. From the record number of the papaersinvited.com server database the query is answered and the data is retrieved to populate the page corresponding to the URL. This HTML page doesn't exist statically but generated dynamically when we put the query on the site specific search and the corresponding script is run on the papaersinvited.com server to retrieve data. Thus, any Web crawler cannot detect the URL unless someone posts it on some static page (link say Yahoo! or Google personal page), then it ceases to be Deep any more. This is called the gray zone [2] where Deep contents may appear at surface at times. But this does not solve the problems arising in Deep Web Search Engines. From the volume of information content listed in [2], it is quite clear that static ranking of all searchable URLs from all Deep Web site databases is next to impossibility. Yet most of the quality data valued by and indeed searched by Web surfaces around the world reside in the Deep Web databases, hidden from common Search Engines. The BrightPlanet listings of 60 major Deep Websites contains valuable data in various fields with a size around 750 terabytes roughly to 40 times that of the known Surface Web data [2]. However traffic to these sites is almost half as that to surface sites. This is primarily because of the fact that they cannot be detected or ranked by conventional Search Engines. Thus it is worthwhile to further look into the impossibility factor of ranking Deep Web pages by the conventional approach to ranking systems. We then propose a novel approach to optimally search and rank Deep Web pages by using a probabilistic model and some optimization techniques in operations research.

The rest of paper is organized as follows. In Section 2, we formally specify the problem and describe a probabilistic model and an optimization technique that form the foundation of our eventual novel approach to Deep Web Search Engines. Section 3 is devoted to the development of our proposed system. Section 4 contains a detailed experimental evaluation of our proposed approach. We conclude in Section 5.

2. Surface Web Search Engine

In this section, we shall briefly review a general concept of Surface Search Engine, in other words conventional Search Engine like AltaVista or Google and the ranking methods generally followed by a Surface Search Engine. The process of development of conventional or Surface Search Engines begins with the process of development of

Web page ranking techniques. Presently the focus of all major Surface Search Engines is towards link based Web ranking techniques [11]. For example, Google's make use of PageRank algorithm [12] but because of some flaws now it has changed its focus towards a better alternative system for link based Web page ranking.

The name PageRank is trademark of Google and it is considered as the heart of Google search process. PageRank is a probability distribution making use of a random surfer model used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank relies on the uniquely democratic nature of the Web by using its vast link structure as an indicator of an individual page's value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages 'important'. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

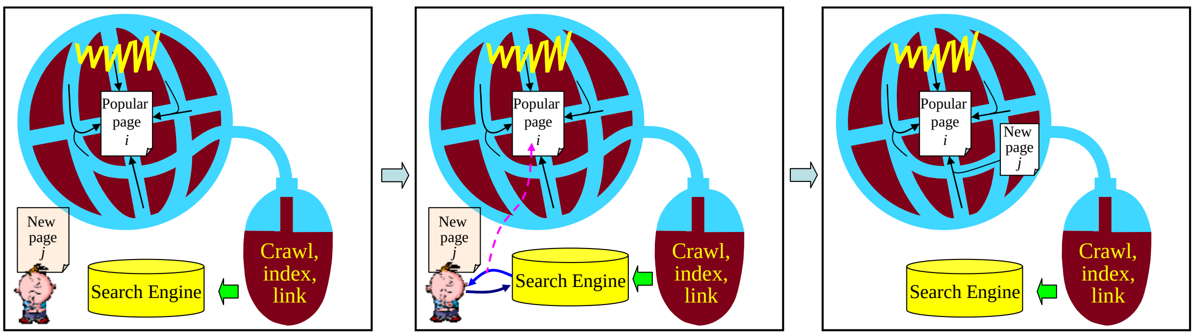
But despite its simplicity, efficiency and huge popularity it is being found that there are certain flaws in the PageRank that leads to certain problems.

(i) Dangling Links
(ii) Historical Values
(iii) Problems with Web Spams- False Links, Google Bombing, Google Jacking, Google Juice.

One common belief is that the use of Search Engines biases traffic toward popular sites. Pages highly ranked by Search Engines are more likely to be discovered and consequently linked to by other pages. This in turn would further increase the popularity and raise the average rank of those pages. As popular pages become more and more popular, new pages are unlikely to be discovered. Such a cycle would accelerate the rich-get-richer dynamics already observed in the Web's network structure and explained by preferential attachment and link copy models [13-15]. This is at the origin of the vicious cycle illustrated in Fig. 2. Fig. 2(a) illustrates that existing Page *i* with high link popularity and high rank while a new Page *j* is created. In Fig.2(b), the new Page creator discovered the information of Page *i* through Search Engine. Then, the Page *j* is linked to Page *i* so that Page *i* becomes even more popular from the Search Engine's perspective. Methods to counteract it are being proposed [16-18].

In [16], H. Hama et al. have proposed a new model for novel Surface Search Engine by introducing a new concept of Popularity Rank (PR) along with an establishment of a Popularity Ranking Operator which is combined with new link structure analysis. In their work, they considered two kinds of relationships: page-to-image and image-to-page, which ultimately lead page-to-page model and image-to-image model. In addition to these two models, they adopted a new concept of popularity ranking which will play an important role in development of Search Engines. Briefly the new concept of Popularity Rank can be restated as follows.

Users want to find as much as popular, famous and familiar images as possible when there are many images which are similar with user's query. For example, suppose that there is an image with "Name: unknown, PR=low". Once the image is known as "Name: *Pooh*", then its PR must become high after re-computation of PR. Because Domain "Walt-Disney" must have high PR, and also "Name: *Pooh*" must be strongly related with the Domain "Walt-Disney". Like this, if the node has some strongly related attributes (for example, same name) with any attributes of at least one high ranked node, then it has high PR without any link as illustrated in Fig.3. Such kind of effects can not be seen when only Link Ranks are used. In Fig.3(c), every node may be HP (Home Page), image, or domain which has links and popularity relations.



(a) (b) (c)

Fig. 2 Current Search Engine tendencies.

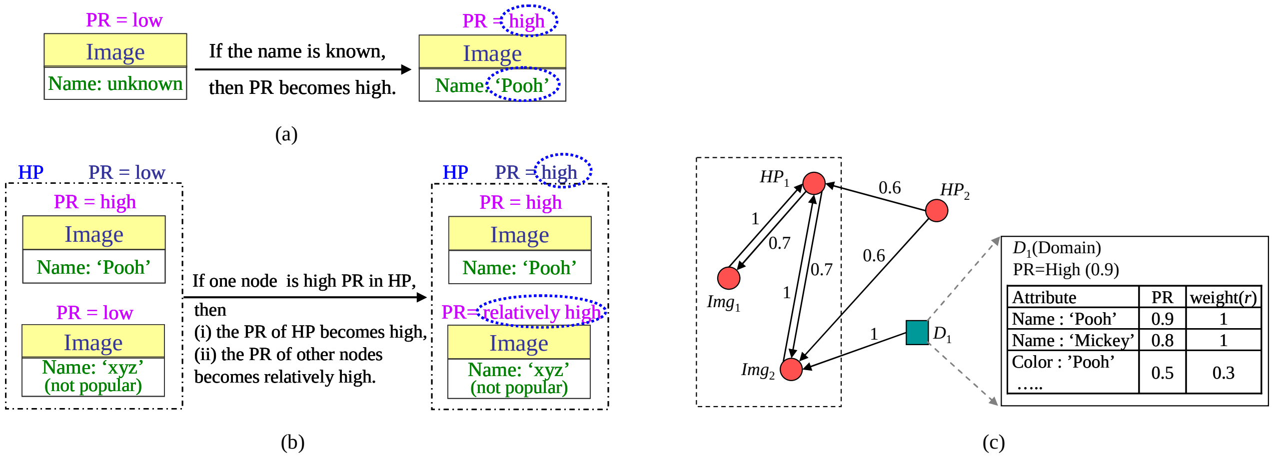


Fig. 3 Popularity as a criterion for node ranking: (a) between images, (b) between HPs, and (c) links among nodes

A significant improved development of Surface Search Engine ranking system was made by our previous work [17]. We explored and examined a novel approach to ranking system based on some special types of Markov chain along with new concepts of popularity and relevancy measures. The proposed ranking system consists of special types Markov models, popularity based models and concepts of relevance. Ranking functions which are key components of image ranking system are first developed and combined in order to develop a new general ranking system. In addition, the combination is reinforced with concepts of relevance models. Briefly, our previous work is as follows.

Consider an image database D with n images $[I_1, I_2, ..., I_n]$. Let Q be a query or set of queries. Then our proposed method develops a ranking system for the images in the image database according to the combined ranking functions obtained by special type Markov model and popularity based model. Let $SM(R)$ and $PM(R)$ denote special Markov and popularity ranking functions respectively. These two ranking functions are to be integrated in order to obtain image ranks for the database. Overview of our ranking system is as shown in Fig. 4 with pictorial representations.

The results with these models stimulate us to do research works which will lead to a new Deep Web Search Engine.

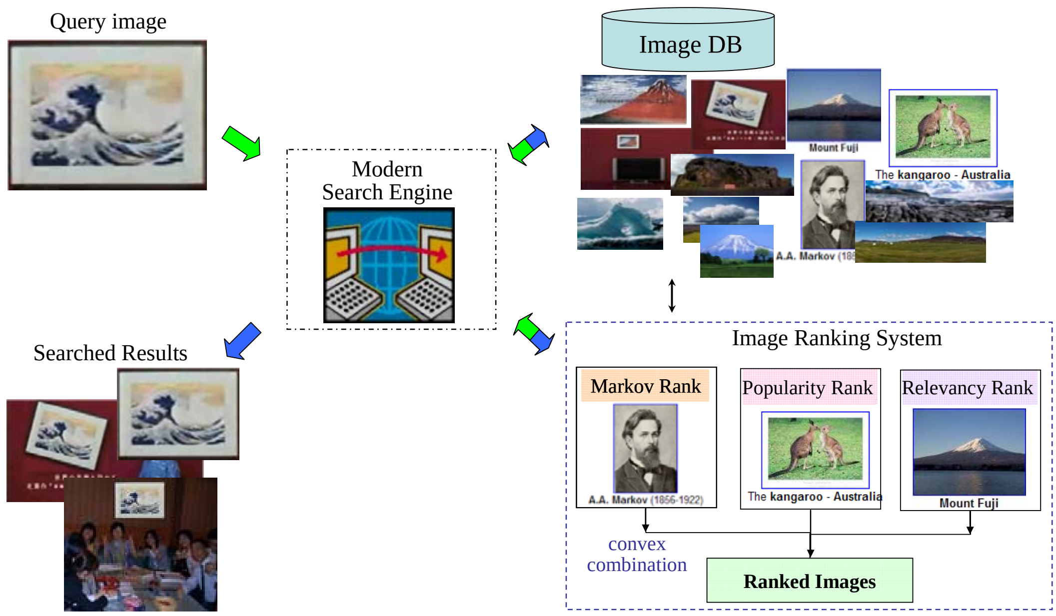


Fig. 4 Overview of ranking system for a modern Surface Web.

Since the Deep Web pages do not exist until a directed query is posted to a site specific database and unless somebody posts an URL like http://... = statistic on some static page, there is no citation for that particular page having the specific Deep Web site database. We will attempt to develop an entirely different approach for Deep Web search in the following sections. The approach will depend on probabilistic estimates of contents for various contexts rather than citation of a particular Deep Web page.

3. Deep Web Search Engine

The most significant challenge is the philosophical difference between the Surface and Deep Webs with respect to how data is stored. In the Surface Web, data is stored in document files including images. But in the Deep Web, data is stored in databases or produced as the result of a computation. This difference is fundamental and implies that traditional document indexing techniques, which have been applied with extraordinary success on the Surface Web, are inappropriate for the Deep Web. Since nothing is sure in Deep Web, probabilistic approach may be appropriate. So we propose a probability model for Deep Web. It is based on probability criteria rather than hyperlink structure which not occurred in Deep Web. The proposed model is based on general multimedia types of keywords such as text, images, audio, etc. For simplicity, we consider here only three types of keywords, namely,

(i) Text-based Keyword (TK-based)
(ii) Image-based Keyword (IK-based)
(iii) Hybrid-based Keyword (HK-based).

We assume that each keyword is linked to a set of Deep Websites. Only we don't know 'how' or 'in what way'. But we do know that these are customized searchable sites from now onwards let us call set of these sites as 'Generators'. For example:

(1) A Generator with heading 'Research' may contain a keyword 'Conference'. The word conference may point to a Deep Web site *http://www.papersinvited.com*. It is possible that a keyword in a particular Generator will point to a number of sites and a keyword may appear in a number of Generators.

(2) A Generator with heading 'Painting' may contain 'Mona Lisa Face' as image-based keyword (IK-based). It may lead to a Deep Web site *http://icom.museum/vlmp/galleries.html*.

The overview of the proposed system can be seen as shown in Fig.5.

Let G_1, G_2, ..., G_n be Generators based on keywords and subjects covered for a number of Deep Websites. This can be done because the Deep Web sources are available.

Suppose k_1, k_2, ..., k_m be a pool of keywords. It may be TK-based or IK-based or HK-based.

p_{ij}= Probability that k_i in G_j points to sites having relevant information asked in a query when k_i belongs to G_j.

Let x_{ij} = number of relevant Websites when keyword k_i is entered into Generator G_j.

m_i = maximum number of Websites for keyword k_i.

n_j = number of Websites which can relate to keyword $k_1, ..., m$.

Thus, the Deep Web search can be formulated as a linear programming problem as follows:

$$\text{Maximize } \sum_i \sum_j p_{ij}\ x_{ij}$$

with constraints

$$\sum_j x_{ij} \le m_i \qquad (i = 1, 2, \cdots, n)$$

$$\sum_i x_{ij} \le n_j \qquad (j = 1, 2, \ldots, m)$$

$$x_{ij} \ge 0.$$

By using the values of x_{ij} and p_{ij}, we can obtain a new ranking system for Deep Web that enable to make an optimal search. These concepts are illustrated by the experimental results is Section 4.

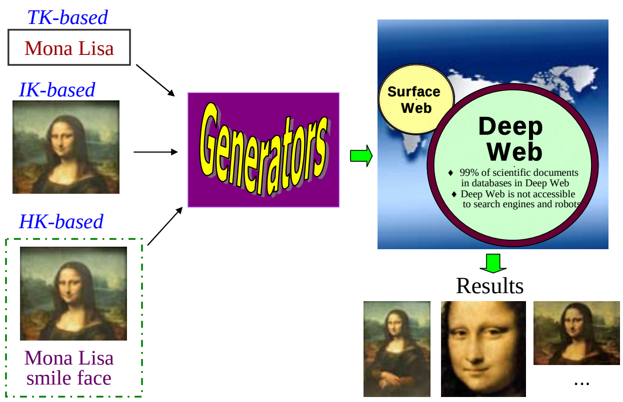


Fig. 5 Overview of Deep Web Search Engine.

4. Experimental Results

In this section, we shall focus on the estimation of the parameters p_{ij}. Three experiments are conducted by using text-based, image-based and hybrid-based keywords. The illustrations are given in Fig. 6 (a), (b) and (c), respectively. For the purpose of explanation, we consider a pool of sampling training set with 5 identified distinct keywords $k_1, k_2, ..., k_5$. We assume three Generators G_1, G_2, G_3 and that each of the keyword belongs to one Generator. The Generators are as given in Table 1(a) that can be obtained from the samples of Generators. A keyword can belong to more than one Generator.

We consider a set of training samples, which are sets of keywords identified from user query search string samples that can be collected. However, generating random samples from hidden databases presents significant challenges. The only view available into these databases is via the proprietary interface that allows only limited access, for example, the owner of the database may place limits on the type of queries that can be posed, or may limit the number of tuples that can be returned, or even charge access costs, and so on. The traditional random sampling techniques that have been developed cannot be easily applied as we do not have full access to the underlying tables. For simplicity, we consider mainly single-table databases with Boolean, categorical or numeric attributes, where the front end interface allows queries in which the user can specify values of ranges on a subset of the attributes, and the system returns a subset (top-*k*) of the matching tuples, either according to a ranking function or arbitrarily, where *k* is a small constant such as 10 or 100. Table 1(b) below gives the sample we will consider in the example.

Now take sample sets of keywords. If a Generator points a relevant site when keyword k_i is entered in the Generator, then we will say k_i is true or will be denoted by k_i(T). Otherwise, it will be denoted by k_i(F). The entries may be seen as shown in Table 1(c).

$$p_{ij} = P\left(k_i \in G_j \text{ points to a relevant site}\right)$$

For example: In G_1, k_1(T) occurs once. Thus, we have

p_{11} = probability that G_1 points relevant site when keyword k_1 is presented.
= 1/5 = 0.2.

Thus, from the sample we have the following probability relation matrix (Table 2). From these numerical tables, we can derive optimal search results by using the linear programming method described in section 3.

We will assume that $m_i = 2$ for all i and $n_j = 3$ for all j. We then have the linear programming problem formulation in the context of transportation problem as follow.

Objective function:

$$\text{Maximize} \quad 0.2x_{11} + \cdots + 0.9x_{33}$$

$$\text{Subject to} \quad \sum_{j=1}^{3} x_{ij} \le 2, \quad \sum_{i=1}^{3} x_{ij} \le 3, \quad x_{ij} \ge 0.$$

By using the standard technique of linear programming problem, we get the optimal solution:

$x_{12} = 1, x_{23} = 1, x_{32} = 1, x_{33} = 1$ and $x_{ij} = 0$ for all other i and j.

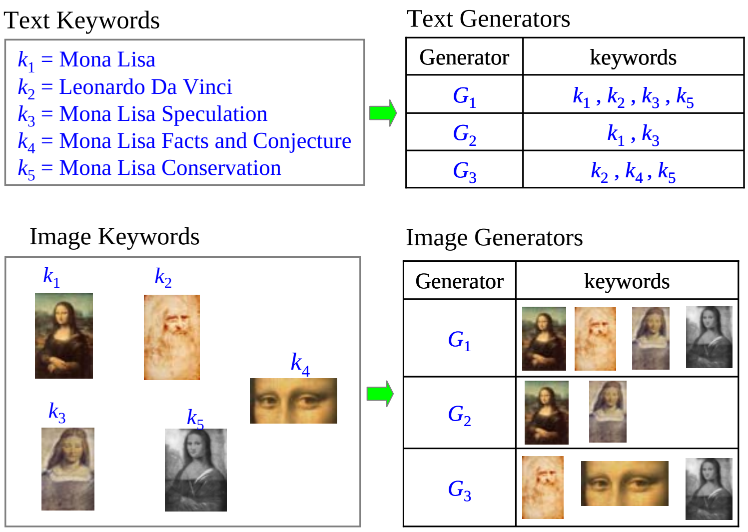


Fig. 6 Illustration of keywords and Generators.

Table 1 Sampling results.

(a) Keyword vs Generator relation

G_1	G_2	G_3
k_1 k_2 k_3 k_5	k_1 k_3	k_2 k_3 k_5

(b) Keywords samples

Sample number	Keyword identified
1	k_1, k_3, k_5
2	k_2, k_3
3	k_4
4	k_2, k_3, k_5
5	k_1, k_4, k_5

(c) Keyword-Generator relevance

Sample	G_1				G_2		G_3		
	k_1	k_2	k_3	k_5	k_1	k_3	k_2	k_3	k_5
1	T	F	F	F	F	T	F	F	F
2	F	T	T	F	F	T	T	T	F
3	F	T	T	F	F	F	F	T	F
4	F	T	F	F	F	T	F	T	F
5	F	F	T	F	T	T	T	T	F

Table 2 Relational Probability matrix.

keyword	G_1	G_2	G_3
k_1	0.2	0.5	0.3
k_2	0.2	0	0.5
k_3	0.6	0.8	0.9
k_4	0	0	0
k_5	0	0	0

This result can be interpreted as follow:

$x_{12} = 1$ => the best result for given query will come out when keyword k_1 is entered into Generator G_2.

$x_{23} = 1$ => $k_2 \in G_3$ gives best results.

$$\left.\begin{array}{l} x_{32} = 1 \\ x_{33} = 1 \end{array}\right\}$$ => $k_3 \in G_2$ and $k_3 \in G_3$ give the best results.

Optimal results can be obtained from the Websites that are pointed by $k_1 \in G_1$, $k_2 \in G_3$, $k_3 \in G_2$ and $k_3 \in G_3$.

We noted that in the example the associated probabilities with $p_{12} = 0.5$, $p_{23} = 0.5$, $p_{32} = 0.8$ and $p_{33} = 0.9$.

Thus, the ranking system can be introduced as shown in following:

Relevant pages pointed by keyword k_3 in G_3 = rank 0.9

Relevant pages pointed by keyword k_3 in G_2 = rank 0.8

Relevant pages pointed by keyword k_1 in G_2 = rank 0.5

Relevant pages pointed by keyword k_2 in G_3 = rank 0.5.

5. Conclusions

While considering the model, one must keep in mind that Deep Websites are valuable only if specific, but detailed and valuable information when a particular topic is searched for. The volume of the information, the number of unique keywords in each of the sites and finally the number of ever increasing population of the Deep Websites makes it practically not very tractable to design a general purpose Search Engine that can cover all the topics and contents of all the Deep Websites. Thus, search techniques for Deep Websites, in order to be efficient must concentrate on specific areas and find associated Deep Websites. In this paper, we have presented a probabilistic model along with an optimization technique for optimal search. Our model uses the results derived from the queries and Generators to calculate the appropriateness of results searched. Moreover, we have used a random sampling technique to gather statistics and experimental results are found based on these sample statistics.

However, due to large range of variations in case of user query string, the sample data for the model in this case for the accuracy of probability estimates may become voluminous. But the simplicity of the calculating the estimates is indeed a remarkable point to note for the model. Further the ranking scheme that has been proposed is supported to perform with high efficiency, at least in theory, overcoming the difficulties faced by normal ranking algorithm, in case of Deep Web Search Engines.

Acknowledgments

This work is partially supported by KAKENHI 19500154 Grant-in-Aid for Scientific Research(C) and Research Fellowships of the Japan Society for the Promotion of Science.

References

[1] Ibrahim A., et al., “Addressing Effective Hidden Web Search Using Iterative Deepening Search and Graph

Theory", *IEEE 8th Intl. Conf. on Computer and Information Technology Workshops*, July 2008, pp. 145-149.
[2] Michael K. Bergman, "The Deep Web: Surfacing Hidden Value", *The Journal of Electronic Publishing*, vol. 7, no. 1, Aug.2001. "http://www.press.umich.edu/jep/07-01/bergman.html".
[3] Peiguang Lin, Yibing Du, Xiaohua Tan, Chao Lv, "Research on Automatic Classification for Deep Web Query Interfaces", *Intl. Symp. on Information Processing* (ISIP), Moscow, pp. 313-317, May 2008.
[4] Wensheng Wu and AnHai Doan, WebIQ, "Learning from the Web to Match Deep-Web Query Interfaces", *Proc. of the 22nd Intl. Conf. on Data Engineering* (ICDE'06), Atlanta, USA, Apr. 2006.
[5] Govind Kabra, et al., "Dewex: An Exploration Facility for Enabling the Deep Web Integration" *Proc. of the 23rd Intl. Conf. on Data Engineering* (ICDE'07). Istanbul, 2007, pp. 1511-1512.
[6] Andrei Broder, et al., "Graph structure in the Web", *The International Journal of Computer and Telecommunications Networking*, vol. 33, pp. 309-320, 2000.
[7] Ping Wu, Ji-Rong Wen, Huan Liu,Wei-Ying Ma, "Query Selection Techniques for Efficient Crawling of Structured Web Sources", *Proc. of the 22nd Intl. Conf. on Data Engineering* (ICDE'06), Atlanta, USA, Apr. 2006.
[8] Luciano Barbosa, Juliana Freire, "Siphoning Hidden- Web Data through Keyword-Based Interfaces", presented at the *Brazilian Symposium on Databases* (SBBD), Oct. 2004.
[9] Yanbo Ru, Ellis Horowitz, "Indexing the invisible web: a survey", *Online Information Review*, vol. 29, no. 3, pp. 249-265, Emerald Group Publishing, 2005.
[10] James Caverlee, Ling Liu, David Buttler, "Probe, cluster, and discover: focused extraction of QA-pagelets from the Deep web", *Proc. of the 20th Intl. Conf. on Data Engineering*, Boston, MA, pp. 103-114, 2004.
[11] Gupta S., Jindal A., "Contrast of link based web ranking techniques", *Intl Symp. Biometrics and Security Technologies* (ISBAST 2008), Apr. 2008, pp. 1-6.
[12] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine", *Proc. of the 7th Intl. World Wide Web Conf.*, 1998.
[13] Barabasi and R. Albert, "Emergence of scaling in random networks", *Science*, vol. 286, no. 5439, pp. 509-512, Oct. 1999.
[14] J. Kleinberg, et al., "The Web as a graph: Measurements, models and methods", LNCS, 1627, pp. 1-18, 1999.
[15] S. Kumar, et al., "Stochastic models for the Web graph". *Proc. of 41st Annual IEEE Symp. on Foundations of Computer Science*, pp. 57-65, 2000.
[16] Hiromitsu Hama, Thi Thi Zin, and Pyke Tin, "Novel Search Engine: Combination of Link and Popularity Rank for Multimedia Retrieval", *Proc. of ICICIC2008*, Dalian, China, Jun. 2008.
[17] Hiromitsu Hama, Thi Thi Zin, and Pyke Tin, "Ranking System for Image Database Using Special Type of Markov Chain", *Digital Image Computing: Techniques and Applications* (DICTA2008), Canberra, Australia, Dec. 2008.
[18] Hiromitsu Hama, Thi Thi Zin, and Pyke Tin, "A Hybrid Ranking of Link and Popularity for Novel Search Engine", *International Journal of Innovative Computing, Information and Control*. (submitted)

Pyke Tin received the B.Sc (Hons.) degree in Mathematics in 1965 from University of Mandalay, Myanmar, the M.Sc degree in Computational Mathematics in 1970 from University of Rangoon, Myanmar and the Ph.D degree in Stochastic processes and their applications in 1976 from Monash University, Australia. He was the Rector of the University of Computer Studies, Yangon and Professor of Computational Mathematics. He is now a visiting Professor of Graduate School of Engineering, Osaka City University, Osaka, Japan. His research interests include image search engines, queueing systems and applications to Computer vision, Stochastic processes and their applications to Image processing.

Thi Thi Zin received the B.Sc (Hons.) degree in Mathematics in 1995 from Yangon University, Yangon, Myanmar and the M.I.Sc degree in Computational Mathematics in 1999 from University of Computer Studies, Yangon, Myanmar. She received the M.E. and the Ph.D. degrees in Information Engineering from Osaka City University, Osaka, Japan, in 2004 and 2007, respectively. She is now a Postdoctoral Research Fellow of Japan Society for the Promotion of Science, in Osaka City University, Japan. Her research interests include ITS, Color Image Processing, Mathematical Morphology and so on. She is a member of IEEE.

Hiromitsu Hama received the Dr. Eng. degrees in Electrical Engineering from Osaka University, Japan. He was with Department of Electrical Engineering, Osaka City University in 1971. He is currently Dean of Graduate School of Engineering at Osaka City University. His research interests are in 3D image processing/understanding, ITS (Intelligent Transportation Systems), Image Indexing, computer vision and so on. He is a member of ITEJ, IEICE, SPIE and IEEE.