Private information protection System with Web-Crawler

Myung Sil Choi*, Hyun Woo Kim§, Yong Hwan Kim*, Kyung Ho Chung*, and Kwang Seon Ahn*
Department of Computer Engineering, Kyungpook National University
*{silchoi, hypnus, mccart, gsahn}@knu.ac.kr, §netproi@paran.com

Abstract

The corporate network environment is evolving to integrate and synchronize with various systems. In particular, corporations can now easily share information via the Internet, and recreate it into a resource of knowledge. However, private information leakage always occurs in the course of handling data over the Web. Thus, in sharing information, we attempt to determine how much private information has been exposed on the corporate websites so as to delete it or block its exposure. This paper presents a framework of enterprise management systems (EMS) to enable a corporation to gather and manage its own information. Our system first searches information on corporate websites. Second, the system basically uses a web-crawler as a search method, but closely searches through websites to look for private information patterns exposed on web pages or in attached files. Our search method is named as EMS-crawler. As a result of experiments, our EMS-crawler can gather information two to three times more than standard crawler methods.

1. Introduction

Most e-commerce enterprises are conducting researches to effectively accumulate and utilize a variety of information through globalization or strategic alliances. For example, each enterprise is conducting research to develop several knowledge management systems or to integrate various systems that effectively work connectedly [1]. Among the existing knowledge management systems, the portal-based systems have attracted a great deal of research attention of late. In this regard, there are the Enterprise Information Portal (EIP) to integrate internal/external information into a Web environment, and the Enterprise Knowledge Portal (EKP) to integrate and manage a range of knowledge items, an instrument for cooperation with each enterprise [2].

Sharing data via the Internet is a frequently used method of integrating data between corporations. However, while sharing information through the Web is convenient, it poses the problem of leaking information. This is because private information listed on corporate computer networks, such as resident numbers, addresses, and mobile phone numbers can be exposed to others. Thus, before beginning the sharing of Webs between corporations, it is important to identify private information listed on one's website.

In this paper, we intend to present a framework designed to extract private information exposed on the Web and determine its quantity. To that end, our framework has the following features.

1) We extract private information exposed on corporate websites and determine its quantity. This requires a close search; thus we use the existing web crawler method as it is, but we extract private information patterns along URLs contained in files which are attached to web pages. A Web Crawler (also called a Robot or a Wanderer) is a program that automatically and recursively traverses a Web site retrieving document and information in the World Wide Web. The most common types of Web robots are the search engine wanderers. These robots visit Web sites and follow the links to add more information to the search engine database. 2) The standard Web-search-system-collecting methods are used, but a detailed method of improving the existing system problems is likewise used. The standard Web search engine has difficulty extracting the required information. It is not easy to search various documents or files in Web pages and to verify which of the information included therein is useful. Especially given that the numbers of Web page surges and Web technologies are enhanced in proportion to the growth of the Internet, information retrieval is becoming more and more difficult. In fact, the existing search systems, such as Google [3] and Yahoo, are exerting efforts to improve their performance (eg., to shorten the cycle of recollecting [4], Parallel Crawler [5], Finding replicated Web Collections [6] and Crawling the hidden Web [7]). In spite of their improved Web search performance, the existing systems have yet to accept and implement the Web technologies that are being developed daily.

Then, why is the standard crawler method unable to gather web documents contained in certain URLs? Main links of recent homepages are frequently comprised of flashes. If multimedia files or binary files exist in the web, the standard web crawler method cannot trace Link URLs; for instance, when Link URLs are connected via the Javascript function, or when Link URLs are contained in contents such as multimedia files or binary files. Our system is designed to be able to search even these Web documents that cannot be gathered via the standard web crawler method, and to set up a regular pattern aimed to easily extract private information contained in web documents.

In this paper, the results of the study that was conducted on the Enterprise Management System (EMS), which collects, analyzes, and summarizes information regarding each enterprise and then sublimates the existing information into new information or knowledge, are presented. It was proven, through a performance study, that the proposed Web Crawler of EMS could provide two to three times more search results than the existing Web search system could.

The rest of this paper is organized as follows: we start in Section 2 with the reviews on Web-Crawler. In Section 3, we introduce the structures and functions of EMS, In Section 4, we present results of performance study and we evaluate the suggested Web Crawler of EMS, and in Section 5 concludes the paper.

2. Web-Crawler

The Robot agent was used to rotate the Web, to collect information from Web pages, and for statistical analysis. In 1993, Matthew Gray of MIT attempted to find out how many Web servers existed then that used 'Worldwide Web Wanderer'. Devised by Brian Pinkerton of Washington D.C. University in the early part of 1994, WebCrawler is a search engine that searches for information and initiates services using the tools of 486 personal computer servers. Robot searches for more than millions of Web sites around the world and stores these in the form of indices.

When a link is found by Robot somewhere along the way, the linked information is collected. At present, the Robot search engine stores the automatically collected information contents in a database. The Robot agent plays another role: that of mirroring the contents of other Web sites. The most widely used mirroring tools are Webcopy and w3mir. These programs function by approaching Web pages and storing them as local files. Robert also functions, though, as a search engine, automatically and periodically navigating Web pages and searching for and collecting data. It plays the role of a medium for receiving keywords between the users as well as varied data and search results. As indicated in the description above, Robot agent carries out the diverse and complex functions, and at the same time, it continues to become enhanced [8]. However, the existing search engine has a particular weakness: private information leakage always occurs in the course of handling data over the Web. In this paper, we attempt to determine how much private information has been exposed on the corporate websites so as to delete it or block its exposure.

3. EMS System Overview

In this section, the EMS structure and its functions of collecting documents and carrying out management on the Web will be discussed.

3.1. Architecture of EMS

Fig. 1 shows the structure of EMS. Our system consists of two steps from the Web Crawler engine (WCE) and a knowledge summary indexing engine (KSIE), respectively: collecting information from WCE and summarizing and maintaining all documents from KSIE.

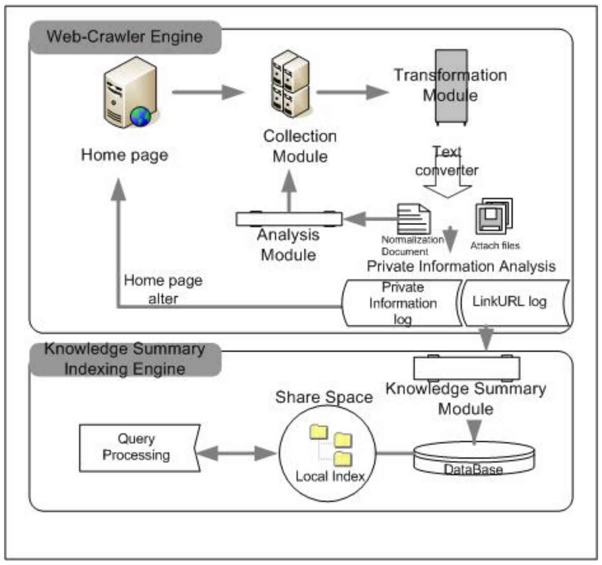

Figure 1. Architecture of EMS

According to the bilateral agreement mentioned in section 1, the Web Crawler engine can access only the Web server of the joint enterprises. The Web server of the relevant enterprises must allow the Web Crawler engine to approach "robot.txt" in accordance with Robot Exclusion Standard [9].

A collection module (CM) approaches the Web server of the enterprise and collects Web documents (including attached files). The attached files are then filtered out to a transformation module (TM), which parses the contents of attached files (ppt, xls, doc, …)

and performs a transformation operation to extract link URLs. Meanwhile, Web pages are saved as normalization documents (NDs) in the form of a tag structure. In the Analysis Module(AM), we analyze Link URLs contained in ND and private information patterns. These private information patterns are based on Java's regular expression.

A knowledge summary indexing engine (KSIE) summarizing and maintaining all web pages (filtering information in WCE) within a website. Especially, Share space maintain local index on the summary information.

3.2. How it works

3.2.1. Web Crawler Engine

The focus of this study was how much information can be obtained from the Web Crawler engine. The function of each component is as follows:

Collection Module(CM) : It functions to collect Web pages. Fig. 2 shows the algorithms for and steps in the collection of Web pages by link URL.

```
Algorithm 1: Collection Process
1. Insert a Web URL of Enterprise in 'Internal URL'
2. for each 'Internal URL list'
3.     Download Web page
4.     Call 'Transformation Module'
5.     Register Web page URL in 'Visited URL'
6      Extract included URLs
7.     Store included URLs to 'Temp URL'
8.     for each 'Temp URL'
9.        if Domain name was same
10.           Insert a Temp URL in 'Internal URL'
11.       else
12.          Insert a Temp URL in 'External URL'
```

Figure 2. Algorithm 1: Collection Process

First, it inserts an enterprise's Web URL to be collected into an internal URL and then downloads Web documents in subsequent order (lines 1-3). Second, the files attached to a Web page are filtered out onto a TM (line 4). Third, the pertinent collected page's URL can be registered as a "visited URL" (line 5), thus preventing the collected documents from being collected in an overlapping way. The included URLs are extracted from the collected Web documents and are saved as "Temp URL" (lines 6-7). When the included URL (i.e., Temp URL) and domain name of a visited URL registered in line 5 are identifiable, "Temp URL" can be inserted in "Internal URL" (lines 8-10). When they are not identifiable, "Temp URL" can be inserted in "External URL" (lines 11-12).

Transformation Module(TM) : The Web document attached files collected in CM are analyzed with the transformation of the text. Multimedia files such as flash and binary files (doc, ppt, xls, and hwp) are used for this purpose.

While the existing Web engines exclusively analyze, parsing Web documents with focus on the text, the proposed system analyzes URLs and data, transforming binary documents through version modules and going through the "parsing" process.

As has been shown, the proposed research method can visit URLs more often than the existing Web search engine can, and it can gather more Web information.

The collected Web page is transformed into a Normalization Document (ND), as shown in Fig. 3. The structures include URLs, attached-file names, and link URLs. When the collection of <<PrimaryUrl:>> is being performed, the relevant Web page's URL is recorded. When the collection of <<AttachUrl:>> is being performed, the link URLs of the attached files are recorded. Here, the URLs are transformed into absolute URLs before being recorded.

```
<<Web DOCUMENT >>
<<PrimaryUrl:>>
http://www.k.....eng/index.jsp?menuID=20070523193053896133
<<Title:>>
.contents_size { width:630px; } .left_color { background:#BCBBB6;
<<Content:>>
Admission is based on the documents submitted, including
recommendations, personal statement, and academic achievement record
f
<<Attach:>>
attach-1111992711.doc
attach2040930794.doc
;
attach36622442.doc
<<AttachUrl:>>
http://www..:8080/common/images/knu_eng/images/down/download1.do
   c
http://www..:8080/common/images/knu_eng/images/down/download1.do
c
        ;
oad4(Recommendation).doc
<<Date:>>
20080318
```

Figure 3. Normalization document

ND also compares whether the name or size of a document is identifiable. If the name or size is identifiable, ND regards it as an overlapped document and prevents it from being gathered. NDs can be saved in a database and can be used for searches to address a query from a general customer.

Analysis Module(AM) : An ND has been cited as the form produced to gather the link URLs included inside it with ease. It continues to parse <<AttachUrl:>> link URLs, follow link URLs, and gather information. Such processes can operate with several threads, adjusting the number of threads to have an effect or influence on an underweighted Web server. It also prevents duplicated URLs from being gathered, especially in the analysis step. It functions as such because continuing to collect a once-visited URL damages the statistical data and changes the accurate page rank owing to the duplicated data. They check whether one URL exists in "Internal URL" or in an IP address. The JavaScript Dynamic URL is transformed into an absolute URL, which checks whether a duplicated URL exists. Attached or multimedia files are transformed into text, going through the analysis step of link URLs.

Moreover, the factor that is crucial for the collection of numerous Web documents in an AM is the designation of a page-depth AM. Page depth means the frequency of clicking on and searching for a link address to find a Web page. It also means the depth of following the first "Internal URL" for the purpose of collecting Web documents. Needless to say, the wider the "depth" becomes, the greater the possibility of gathering much information, and the wider the "depth" becomes, the greater the time required to gather it. The Web site features must also be considered and determined.

3.2.2. Knowledge Summary Indexing Engine

Summarization is a necessary step for efficient searching, especially when the amount of information is very large. A summary is very compact representation. Our summarization process consists of two steps by techniques of Vector Space Model (VSM) and Latent Semantic Indexing, respectively. Briefly, in VSM, documents and queries are represented by vectors of weighted term frequences. Three factors may be used in term weighting, i.e., the term frequency (TF), the inverse document frequency (IDF), and the normalization factor. TF represents how frequently a term appears in a document, IDF represents how frequently the term also appears in other documents, while the normalization factor is used to reduce the side-effect of different document sizes on weights.
TF * IDF is the most frequently used equation for calculating weights, which means that a term is important only if it can differentiate a document from others. Similarity comparisons among documents and/or between documents and queries are made via the similarity between two vectors, such as the dot product of two vectors.

Latent Semantic Indexing (LSI) [10, 11] has been proposed to overcome synonymy, polysemy and noise problems in information retrieval. LSI discovers the underlying semantic correlation among documents by building a concept space. A technique known as Singular Value Decomposition (SVD) is used to reduce this concept space into a much lower dimensionality, reflecting the major associative pattern in documents, while ignoring the smaller and less important influences. For an $m * n$ term-by-document matrix A (m equals the number of terms and n equals the number of documents), with the rank (number of vectors in the basis of the column space spanned by the column vectors) of A is equal to r, the SVD is defined as follows:

$$A = U\Sigma V^T,$$

where the matrices U and V are orthogonal matrices ($U^TU=I_m$ and $V^TV=I_n$, where I is the identity matrix).
Σ is a diagonal matrix (i.e., Σ =diagonal($\sigma_1, \sigma_2, .., \sigma_p$), where p=min(m, n), with the remaining matrix cells all zeros) [12].

This summarization step reduces a very high demensional space (of tens of thousands) to a much smaller one (of less than two hundreds) to facilitate indexing in share space. Since the number of summaries may be large, to further improve the efficiency of the system, we maintain indexes on the summary information.

Also note that there is no restriction on the index methods to be used and any index method (e.g., hash table, index trees) can be used. In fact, Our framework is general enough to allow each corporate to autonomously deploy their preferred indexes.

4. Experiment and Implementation

In this Section, we conduct a simulated experiment to determine how much information the proposed EMS can gather. Actually, there are no other frameworks to be compared with our system. However, to show how often our EMS visits Link URLs, we intend to compare it with Google.

4.1. Experiment setup

In the experiment environment, we assume a corporate Web site as "http://www.ce.knu.ac.kr" site,

Page Depth	Number of Link URLs	Number of Attach Files	Number of NDs	Operation Time (sec)
Depth-1	48	1,102	42	27
Depth-5	11,994	1,102	9,370	18725
Depth-10	13,742	1,117	9,448	23500
Depth-20	13,791	1,158	9,526	24675

Table 2. Valuation in accordance with Page depth

then intend to gather Web documents.

The EMS System Web Crawler engine demonstrates the Java language. The settings of the first experiment that was conducted are shown in Table 1.

Set variables	Default values
Attach File type	All type
Collection time	UNLIMIT
The Number of Thread	5
Thread check time	500 ms
Thread time out	600000 ms

Table 1. Initial settings

The collected attached files were designated as different file types (e.g., ppt, pdf, doc, hwp, zip, xls, etc.). The collection time was unlimited until the collection depth was over. The number of threads was found to reach five, at random, for the purpose of maximizing the period of the Web server Web access. The thread check time was found to reach 500 ms, and the thread timeout was found to reach 600,000 ms.

4.2. Experimental Evaluation

As discussed above, there are no frameworks to be compared with our system, but, in order to ascertain how many Link URLs on websites our EMS-Crawler can trace, we intend to compare it with Google for the convenience's sake. To know the number of Web documents on Google, we can use the command "site:".

The objective of the experiment was to compare and evaluate the amount of data obtained by Google and the EMS Web Crawler engine (called EMS-Crawler) from "site:www.ce.knu.ac.kr". Focusing on the collection of as many kinds of Web documents as possible, the proposed system set a Web page depth restriction of 20. Then, the number of link URLs collected was compared with that collected by Google.

Fig. 4 shows the numbers of link URLs collected by Google and EMS Crawler in May 2007 and March 2008, respectively. There is a big difference between the numbers of Google search results in 2007 and 2008, indicating that the past collected information or the annually increasing search results are in accordance with the site features themselves instead of the real-time and present number of search results.

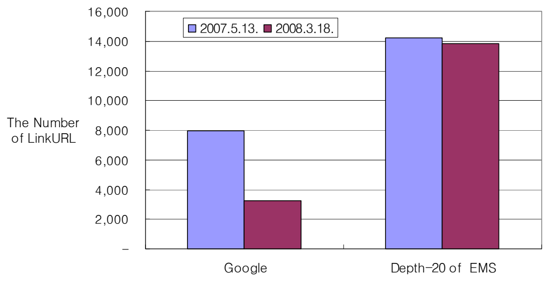

Figure 4. Comparison of the numbers of link URLs in a special site

As described in section 3, the numbers of documents collected through the EMS Web Crawler engine's CM, TM, and AM processes in 2007 and 2008 are slightly different. The numbers, however, are greater than those of Google. The EMS Crawler page depth amounted to 20 in May 2007, but it reached more than 1.78 times that of Google in May 2007 and more than 4.28 times the collection in March 2008.

Table 2 indicates the valuation of the number of link URLs in Web pages, the number of attached files, the collection time, and the number of NDs when the EMS Crawler page depth was 1, 5, 19, and 20, respectively. The focus was placed on the number of diverse URLs collected rather than on the collection speed, as part of a bid to evaluate the performance of the search engine. As indicated previously, the proposed system focuses on the collection of a wide variety of documents.

Fig. 5 shows the EMS-Crawler page depths of 1, 5, 10, and 20 as well as the results of their comparison with the number of link URLs and NDs (N[linkURL]).

Depth 1 was cited as Website pages. The number of link URLs reached only 48, and the included URLs were not searched. The number of NDs that were collected shows the sharply rising number between depths 1 and 5. There was only a slight difference between depths 10 and 20.

As an evaluation subject, N[linkURL] links not only Web pages but also dynamic URLs. It is thus conspicuous when depths 5 and 10 are indicated in comparison with N[ND].

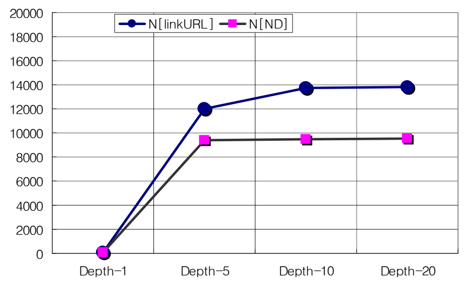

Figure 5. Collection numbers in accordance with the Page-depth

5. Conclusion

In this paper, we proposed a framework to extract private information and confidential information listed in a corporate Website that should not be leaked, and to determine the quantity of such information.
This provides foundations for ensuring the safety in sharing information between corporations. Our system structure does not simply follow the standard web crawler method, but focuses on searching private information contained in web documents. Thus, this requires a closer search method; EMS-Crawler gathers even Link URLs contained in dynamic URLs or binary files of multimedia. As a result, it was found that our proposed method can gather twice or three times as much information as the standard crawler method can.

Our system is designed to be able to search even Web documents that cannot be gathered via the standard web crawler method, and to set up a regular pattern aimed to easily extract private information contained in web documents.

Therefore, our research enables corporations to easily control or block webpages containing gathered private information, and furthermore to provide other corporations with web documents and files from which private information is eliminated. The EMS Web Crawler engine can transform the collected Web pages into NDs and can store these in a database. As such, the collected information can be extended as inter-corporation sharing system.

A KSIE has a knowledge summary module (KSM). The KSM uses and classifies the collected information and reproduces and recommends knowledge or information regarding each enterprise. It is suggested that in the future, a study be conducted on an effective method for the recommendation of knowledge or information.

References

[1] S. Staab, R. Studer and H-P. Schnurr, Y. Sure. Knowledge Processes and Ontologies. *IEEE Intelligent systems & their applications*, Vol. 16, No.1, pp. 26-35, 2001.

[2] S. Lee, Y. Ahn and Y. Min. Design and Implementation of KMS Based on EKP for Effective Knowledge Sharing. In *Proceedings of Knowledge Information management society*, Korea, 2003.

[3] S. Brin and L. Page. The Anotomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World Wide Web Conference*, Brisbane, Australia, 1998.

[4] J. Cho and H. Garcia-Molina. The Evolution of the Web and Implications for and Incremental Crawler. In *Proceedings of the 26th International Conference on Very Large Databases*, Cairo, Egypt, 2000.

[5] J. Cho and H. Garcia-Molina. Parallel Crawler. In *Proceedings of the 11th International World Wide Web Conference*, Hawaii, USA, 2002.

[6] J. Cho, N. Shivakumar and H. Garcia-Molina. Finding Replicated Web Collections. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, Dallas, Texas, 2000.

[7] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. In *Proceedings of the 27th International Conference on Very Large Databases*, Rome, Italy, 2001.

[8] M. Koster. Robots in the Web: threat or treat ?. *ConneXions*, Volume 9, No. 4, April 1995.

[9] M. Koster. A Method of Web Robots Control. Network Working Group, Internet Draft, Dec. 1996, http://www.robotstxt.org/wc/norobots-rfc.html.

[10] M. W. Berry and R. D. Fierro, Low-rank Orthogonal Decompositions for Information Retrieval Applications. *Numerical Linear Algebra with Applications*, Vol. 3, pp. 301-328, 1996.

[11] T. G. Kolda and D. P. O'Leary, A Semi-discrete Matrix Decomposition for Latent Semantic Indexing in Information retrieval. *ACM transactions on Information Systems*, 16, pp. 322-346, 1998.

[12] G. Golub, C. V. Loan, Matrix Computations, third ed., *Johns Hopkins University Press*, Baltimore, MD, 1996