Building a multi-level database for efficient information retrieval: A framework definition.

Spiridon C. Denaxas and Christos Tjortjis
Information Systems Group, School of Informatics
University of Manchester, PO Box 88, Manchester, M60 1QD, UK
S.Denaxas@postgrad.manchester.ac.uk | christos@co.umist.ac.uk

ABSTRACT

With the explosive growth of the Internet and the World Wide Web, the amount of information available online is growing in an exponential manner. As the amount of information online constantly increases, it is becoming increasingly difficult and resource demanding to search and locate information in an efficient manner. Information overload has become a pressing research problem since current searching mechanisms, such as conventional search engines, suffer from both low-precision and low-recall. It is clear that a more dynamic, scalable and accurate searching methodology needs to be developed to overcome these limitations.

This paper proposes a methodology consisting of an amalgamation of several research areas such as Web mining and relational database systems. We develop a proof of concept prototype which consists of an agent used to extract information from individual Web pages and a dynamic multi-level relational schema to encapsulate this information for later processing. The prototype provides users with a higher level of scalability and flexibility and can be utilized for searching the Internet and Intranets across large-scale organizations.

KEY WORDS
Information retrieval, World Wide Web, Semantic Retrieval, Internet, Intranet.

1. Introduction

The World Wide Web is potentially the world's largest knowledge base and can be viewed as one huge, diverse, hybrid and dynamically distributed database. A recent survey [1] revealed that the information available online doubles every 18 months whereas the number of personal homepages doubles every six months. Text and hypertext are used for digital libraries, personal homepages, reviews, news casts, product catalogues, newsgroups, scientific and academic articles and medical reports for individuals, organizations and projects.

The majority of documents that are available online can be characterized as heterogeneous: they do not conform to a consistent standard or style of authorship, nor such standard exists [2]. Web documents are authored by a diverse set of individuals sharing different technical and cultural backgrounds. Thus, the total volume of heterogeneous text and hypertext data greatly exceeds that of structured data.

Searching this vast volume of semi-structured information that is available online poses a significant challenge since it is more sophisticated and dynamic than the information that any current database architecture can store and manipulate. Taken as a whole, the set of Web pages available lacks a unifying structure and presents a far more complex authoring, style and context variation than witnessed in traditional text archives [2]. This increased level of complexity renders “off-the-shelf” database management, information retrieval and information searching solutions practically impossible to use. It is clear that searching the Internet in an efficient manner is a challenging field in which research needs to be undertaken. This vast volume of accessible information online has raised many new opportunities and challenges for knowledge discovery and data engineering researchers.

In this paper, we provide an overview of the current problems associated with the online information searching domain, the current trends and existing solutions and offer a detailed framework specification of a prototype designed to overcome of these limitation and provide users with the ability to perform more accurate and complex queries.

The paper's structure is as follows: In section 2 we discuss the dominant methods for locating information online and illustrate their current limitations. Section 3 depicts the proposed methodology and a detailed analysis of its core components is provided. A prototype system is discussed and assessed in section 4 and finally conclusions and directions for future work are provided in section 5.

2. Background Issues

Users obtain information on the Internet or individual Intranets using two dominant procedures [3]:
a) manually browsing b) utilizing a Web search engines

These procedures are briefly discussed and evaluated below.

2.1 Manual browsing

Browsing refers to the act of following hyperlinks between Web sites and traversing through them until the information requested is located. This process is very often tentative and unsatisfactory since a user might be forced to spend large amounts of time in order to successfully locate the information requested. Additionally, a recent survey [4] revealed that the deep Web holds approximately 500 billion Web pages in it and that public information hidden in it is 400 to 500 times larger than what users can access through the surface Web. This hidden information is only accessible by performing intelligent queries to various distributed database systems. It becomes clear that given the Internet's ever-growing nature, manual browsing will soon become an extremely resource inefficient procedure for locating information online.

2.2 Search engines

The second dominant method for locating information online is by using Web search engines. According to the manner in which information is collected and indexed, Web search engines can be divided into three main categories: directory search engines, crawler-based search engines and meta-search engines [3].
Directory search engines are engines which consist of thematic directories a user can browse and locate information. These directories are manually built and only represent a very small percentage of the Word Wide Web since their respective database cannot be updated in real time to synchronize with the Internets volatile nature. Crawler-based search engines are engines which make use of correlated programs called Web spiders in order to traverse an individual Web site and its respective pages analyze their content and finally add them to a large index database automatically. Finally, meta-search engines are engines that make use of various ranking algorithms such as PageRank [3],[5],[6] or HITS [7] in order to calculate some degree of authority in the results obtained, and effectively rank them before returning them to the users.

Conventional search engines have a number of problems and inherent limitations associated with them. Given the World Wide Web's volatile and expanding nature, most search engines have a very limited coverage of the Internet. One could argue that it is becoming practically impossible to maintain an up-to-date index of the Internet for searching. Additionally, conventional search engines offer little to none interaction with the user: a user cannot make detailed specifications apart from entering a number of keywords.

Search engines are also known [8] to suffer from poor accuracy: they have both low recall (fraction of relevant documents that are retrieved) and low precision (fraction of retrieved documents that are relevant).
A typical search would return a very large amount of low-accuracy results which the user has to manually browse in order to locate the information requested.

Common text searching issues such as synonymy, polysemy (occurring when a word has more than one meaning) and context sensitivity also become severe on the Web [2],[14]. Finally search engines are also limited by various secondary factors such as processing delays and bandwidth bottlenecks.

It becomes clear that existing solutions for locating information on the Internet and individual Intranets have significant deficiencies with respect to robustness, flexibility and precision [9].

3. Proposed Methodology

The proposed methodology consists of two individual components developed separately: the agent and the underlying relational database. The former is responsible for identifying and extracting information from individual Web pages it traverses whereas the latter is concerned with storing and manipulating these data.

3.1 Agent

The agent is responsible for recursively traversing individual Web sites and the respective Web pages that are linked to them. While traversing a Web page, the agent identifies and extracts a set of predefined elements, processes them and forwards the final set of data to the underlying relational database.

The agent extracts both *semantic* and *descriptive* information from the individual Web pages it recursively traverses. Semantic information can be defined as the elements that capture an individual Web pages contents, such as the number of frame elements, ordered or unordered lists and image files on it. The agent is responsible for correctly identifying the HTML meta-tags composing each Web page, filtering out any invalid or erroneous elements and enumerating each of the predefined document elements. The agent processes data by performing a number of sequential conceptual "passes". The elements the agent component is responsible for extracting along with their respective HTML meta-tags are summarized in table 3.1.

The agent heavily relies on proper syntax of the Hyper Text Markup Language it encounters. Due to the Web's hybrid nature, syntax errors and inconsistencies due to different coding styles are bound to exist. While these will not generate fatal errors, a probability that inconsistent or incorrect data will be inserted into the relational database does exist.

Document element	HTML meta-tag
Image file	<IMG SRC= “path”>
Hyperlink	<A HREF = “path”>
Table	<TABLE>
Form	<FORM>
Frame	<FRAME SRC=”path”>
Ordered List	<OL>
Unordered List	<UL>
Radio button	<INPUT type=”radio” NAME=”example”>
Checkbox	<INPUT type= “checkbox” NAME="example">

Table 3.1: The predefined document elements the agent identifies.

Additionally, the agent will further process the number of image media files it identifies on an individual Web page and enumerates their respective types and sizes on a separate process. The types currently supported from the agent component are summarized in table 3.2.

Image File Description	Extension
Monochrome, 16 color, 256 color and 24-bit Bitmap image files	Bmp, dib
JPEG image file	Jpg, jpeg, jpe, jfif
GIF image file	Gif
TIFF image file	tif, tiff
PNG image file	Png

Table 3.2: The supported image formats.

We define descriptive information as the set of information the agent will extract which will provide the user with an abstraction of the individual Web page concerned. Descriptive information includes but is not limited to the documents author, the character set and codepage used or the Web pages title and language. The descriptive information the agent component is responsible for extracting is summarised in table 3.3.

Field	Description
category	The document’s thematic category.
rank	Specifies the document’s rank.
author	The document’s author.
title	The document’s title.
timestamp	Denotes the time the document was inserted into the system.
charset	The document’s character set.
content	The document’s content type.
language	The document’s language code.

Table 3.3: Descriptive information extracted by the agent.

3.2 Filtering

To avoid data duplication and promote data consistency and accuracy, filtering functions are included within the core of the agent component. The agent downloads the source code of a particular Web page and scans it for the set of predefined elements. Additionally, it also locates and extracts hyperlinks pointing to other pages and places them in a queue; the individual pages are considered subsequently in queue order resulting in a breath fist search. Links pointing to external Web sites, advertisements or other non-relative information are rejected. Finally, a *visited URL* hash exists which prevents the agent from re-visiting the same Web page.

3.3 Relational database

The relational database component is responsible for storing and manipulating the information the agent identifies and extracts from the individual Web pages it traverses. A dynamic schema was developed in order to cope with the Web’s hybrid and dynamic nature. Essentially, the relational database does not hold complete Web pages, something extremely resource demanding and unrealistic, but instead stores document *abstractions* which encapsulate all the necessary information for latter processing and querying.

The database is organized in a number of logical levels, each of which holds a different type of information concerning an individual Web page the agent component has processed:

- **Level 0:** The World Wide Web it self, un-altered. Although outside the context of the system, it provides the essential base on which the next logical levels are built upon.

- **Level 1:** The semantic information the agent extracts from the individual Web pages it traverses. Layer 1 also contains a more detailed analysis of the image media types the agent identifies.

- **Level 2:** This level contains the descriptive information the agent extracts from the individual Web pages it traverses.

Following the Object Exchange Model (OEM) [10] paradigm, each document abstraction is assigned with a unique Object Identifier (OID) value automatically from the relational database management system. This will effectively form the binding between the information concerning a particular Web page as it is distributed amongst the different levels of the relational database.

Finally, a number of predefined default values are used when the agent fails to locate or extract an element from a Web page; this promotes data integrity and consistency and minimizes the need for data pre-processing when applying data mining algorithms to the data harvested.

4. A prototype system

In order to demonstrate the proposed framework's feasibility and to perform some initial experiments, a prototype system was developed equipped with the majority of the features mentioned in this paper. As this paper presents ongoing work, the systems specifications constantly evolve and new specifications are defined.

The Web site chosen to be processed is the University of Manchester: Institute of Science and Technology (UMIST) [11] homepage and an internal limit of 500 Web pages was set in order to truncate the output. The total time taken to process the sample was: 02':01" and a total of *434* records were created. Out of the 434 records processed, *430* were unique which translates to a mere 0.91% of duplicate information. The remaining records where lost due to throttling occurring from the HTTP server.

The results obtained are summarized in tables 4.1 and 4.2:

Element	Total number
Images	28057
Tables	6194
Forms	446
HTTP links	25017
Frames	3
Ordered lists	0
Unordered lists	13
Checkboxes	6
Radio buttons	0

Table 4.1: The total number of elements processed by the agent.

Data Transfer	Download	Upload
Total data transferred	10.90 MB	0.45 MB
Maximum transfer rate	236.0 kB/sec	9.9 kB/sec
Average transfer rate	92.3 kB/sec	3.8 kB/sec

Table 4.2: The results in terms of *data transfer*.

The total size of the relational database datafiles after the experiment is 1.28MB (1,352,126 bytes) which is relatively small in contrast with the 11MB of data the agent downloaded while processing the sample.

The data harvested by the agent can now be queried using standard relational database query languages like SQL. Additionally, one of the strengths of the defined framework is its compatibility with existing data mining tools and algorithms with no need of extensive preprocessing; the default values used by the system ensure data within the relational database remains consistent. Using clustering, which aims in grouping records together based on their similarity, hidden but potentially useful patterns can be discovered and utilized in the demanding field of Web content mining [12].

4.1 Case studies

In order to fully depict the defined framework's usability and flexibility, a number of case studies were constructed illustrating the different queries that can be performed on the data harvested by the agent.

4.1.1 First case study

A user is compiling a report on various statistical data found on a Web site; the majority of the desired numerical data on that particular Web site is located within table elements. He/she does not require image files of any type or any other data in the form of text. By making use of the fields provided within the relational database, the user is able to specify the minimum number of table elements a Web page should have when the results are returned. The system provides the user with the flexibility to exactly specify the document elements he/she wishes to receive. Furthermore, should the user require information that is of a certain *age*, he/she can additionally specify the minimum timestamp a document must have before it is returned.

4.1.2 Second case study

A user is compiling a medical report on a particular skin disease and wishes to obtain a number of images. He does not require any type of numerical or textual data of any form. Additionally, having to cope with a number of standards, he only wishes to locate JPG type images whose size does not exceed 150KB. By utilizing the fields defined within the relational database component, the user is able to exactly specify the minimum number of images, their respective formats and sizes he wishes to receive. The prototype system will use a standard query language to search the data harvested by the agent component and return it to the user.

It becomes clear by the case studies discussed above that the defined framework provides the user with a much greater level of flexibility than conventional and existing search methods. Additionally, the proposed framework can be used to efficiently locate information on a corporate Intranet, giving users the ability to specify more advanced query parameters than conventional search methods.

5. Conclusion and future work

The amount of available information online is exponentially growing, making efficient information searching and retrieval mechanisms a pressing and research-demanding issue. Existing techniques have significant deficiencies with respect to robustness, flexibility and precision. The current information location trends, such as manual browsing and conventional search engines were analyzed and their shortcomings discussed. This paper investigated the area of locating, identifying and extracting information available online with the ultimate purpose of defining a system framework which will eventually enable more sophisticated, advanced and accurate queries to take place.

One could claim that this work only covers a rather narrow scope of the information searching and knowledge discovery domains. However, the main contribution of this paper is the definition of a framework which can act as the foundation on which more complex and advanced systems can be built upon. The components themselves can be integrated into larger systems and effectively assist users into locating the requested information in an accurate and efficient manner. Finally, data mining algorithms, such as clustering and association analysis, can be performed on the data harvested by the agent component in order to reveal potentially useful but hidden information on the basis of Web mining.

5.1 Future Work

We consider the following various alternatives in order to enhance the proposed framework.

a) Automatic classification of documents.

In order to provide the user with a greater level of flexibility, a thematic classification algorithm could be integrated to the agent and utilized while traversing Web pages. Documents can be classified into several key-topic categories such as *sports, economy-related, academic, leisure, personal* etc.. Users would be able to specify the category they wish to receive results from thus effectively narrowing down the amount of irrelevant topics returned [13].

b) Ranking

Additionally, a ranking algorithm could be applied on the results before they are returned to the user. By utilizing a ranking algorithm, such as PageRank, a certain degree of authority would be calculated and displayed to the user. This would enable him to locate the desired information in a more efficient manner while increasing the relevant quality of the information he receives from the system.

References:

[1] C. Yang, J. Hen, & H. Chen, Intelligent internet searching agent based on hybrid simulated annealing, *Elsevier Decision Support Systems 28,* 2000, 269-277.

[2] S. Chakrabarti, Data mining for hypertext: A tutorial survey, *SIGKDD Explorations 1(2),* 2000, 1-11.

[3] Y. Li, X. Chen & B. Yang, Research on Web mining-based intelligent search engine, *Proc 1st IEEE Conf. on Machine Learning and Cybernetics,* Beijing, 2002, 386-390.

[4] T. Ghanem, W. Aref, Databases deepen the Web, *IEEE Computer*, 2004, 116-117.

[5] C. Hsinchun, C. Yi-Ming, M. Ramsey, C. Yang, An intelligent personal spider (agent) for dynamic Internet / Intranet searching, *Elsevier Decision Support Systems 23*, 1998, 41-58.

[6] S. Brin, L. Page, The anatomy of large-scale hypertextual Web search engine, *Elsevier Computer Networks and ISDN Systems 30(1-7),* 1997, FP11.

[7] L. Longzhauang, Y. Shang, W. Zhang, Link analysis, improvement of HITS-based algorithms on Web documents, *Proc. 11a*