An Information Retrieval Model from World Wide Web based on Formal Concept Analysis

Minyar Sassi Hidri and Amel Grissa Touzi
National Engineering School of Tunis, BP. 37, le Belvédère 1002 Tunis, Tunisia

Abstract: *The World Wide Web (WWW) is a popular and interactive medium to disseminate information today. The Web is huge diverse, and dynamic and thus raises the scalability, multimedia data, and temporal issues respectively. However, the limited storage capacity, the problem of data knowledge extraction on the Web and the need for search for relevant information in a Web site are as many elements which require the recourse to other more advanced techniques. Web Mining can be defined as the extraction of interesting and potentially useful and implicit information from WWW. In this paper, we present an information retrieval (IR) model making it possible to a net surfer to quickly find relevant information from the WWW based on Formal Concept Analysis in both crawler and searcher steps.*

Keywords: *World Wide Web, Data Mining, Information Retrieval, Web Mining, Formal Concept Analysis.*

Received December 26, 2010; Accepted February 21, 2012

1. Introduction

Nowadays, companies, organizations or individuals are submerged by the available information and documents on the Web. Information technologies facilitate its displacement and storage and contribute to their exponential growth. They are in a disproportionate number compared to the human means to treat them. It creates a need for techniques to analyze and extract knowledge from them.

Within the framework of Knowledge Discovery in Data bases (KDD) [11], we have recourse to Data Mining methods [9, 8] to facilitate the hidden data knowledge extraction in the available Web documents. Consequently, the Web and Data Mining should naturally meet. Indeed, the first generates a large flow of information which the second is able to treat and to extract some knowledge from them [5,12].

The development of the Web involved during these last year's an explosion of the data related to its activity.

The set of the used applications to excavate the data from Web with an aim of knowledge extraction bears the name of Web Mining [3,4].

According to analyzed data type, we distinguish three categories from Web Mining applications [1,10]:

- *Web content mining*: for the contents of the Web pages.
- *Web structure mining*: for the structure of the Web sites.
- *Web usage mining*: for the accesses of the users on one or more Web sites.

Data Mining is ensured by means of several methods, we essentially mention Formal Concepts Analysis (FCA) [13].

In this method, a concept represents a couple (Objects, Properties) related by a binary relation. The correspondence between objects and properties is made in the form of formal context. The set of concepts forms a diagram which it called: concepts lattice [7, 6].

The goal of this work is to present a new information retrieval (IR) model using FCA.

The rest of this paper is organized as follows. Section 2 presents a state of the art. We present the principle of search engine, the Web mining and the terminology of FCA. Section 3 presents problems and limits of Web mining. Section 4 present the new IR model on the WWW. Section 5 presents the implementation of this model. Section 6 evaluates the proposed model and section 7 concludes the paper and presents future works.

2. State of the Art

2.1. Search Engines

Information retrieval (IR) has the primary goals of indexing text and searching for useful documents in a collection. Nowadays, research in IR includes modelling, document classification, user interfaces, data visualization, filtering, etc. The task that can be considered to be an instance of Web mining is Web document classification, which could be used for indexing. Viewed in this respect, Web mining is part of the IR process on the Web.

This process is realized using search engines. They are tools which designed to search for information on the

World Wide Web. They are charged to index Web pages in order to allow a research using keywords in a research form.
They work by storing information about many Web pages, which they retrieve from the html itself. They operate in the following order:

- Web crawling
- Indexing
- Searching

Web search engines work by storing information about many web pages, which they retrieve from the html itself. These pages are retrieved by a Web crawler (sometimes also known as a spider) — an automated Web browser which follows every link on the site. Exclusions can be made by the use of robots.txt. The contents of each page are then analyzed to determine how it should be indexed (for example, words are extracted from the titles, headings, or special fields called Meta tags). Data about web pages are stored in an index database for use in later queries. Some search engines, such as Google, store all or part of the source page (referred to as a cache) as well as information about the web pages, whereas others, such as AltaVista, store every word of every page they find. This cached page always holds the actual search text since it is the one that was actually indexed, so it can be very useful when the content of the current page has been updated and the search terms are no longer in it. This problem might be considered to be a mild form of linkrot, and Google's handling of it increases usability by satisfying user expectations that the search terms will be on the returned webpage. This satisfies the principle of least astonishment since the user normally expects the search terms to be on the returned pages. Increased search relevance makes these cached pages very useful, even beyond the fact that they may contain data that may no longer be available elsewhere.

When a user enters a query into a search engine (typically by using key words), the engine examines its index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text. Most search engines support the use of the boolean operators AND, OR and NOT to further specify the search query. Some search engines provide an advanced feature called proximity search which allows users to define the distance between keywords.

The usefulness of a search engine depends on the relevance of the result set it gives back. While there may be millions of web pages that include a particular word or phrase, some pages may be more relevant, popular, or authoritative than others. Most search engines employ methods to rank the results to provide the "best" results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another. The methods also change over time as Internet usage changes and new techniques evolve.

However, proposed search tools, have the following limits. We mention:
The low precision which is due to the irrelevance of many of the search results. This results in a difficulty finding the relevant information,

The low recall which is due to the inability to index all the information available on the Web. This results in a difficulty finding the unindexed information that is relevant.

2.2. Data Mining applied to Internet: the Web Mining

The application of Data Mining methods to the Internet is known under the name of Web Mining [4, 3]. Generally, Web Mining can be defined like the discovery and the analysis of useful information through the Web.

It is defined also as being the application of Data Mining methods to the contents, the structure and the Web resources [10, 1].

We categorize Web Mining into three areas of interest based on which part of the Web to mine: Web content mining, Web structure mining, and Web usage mining [5].

Web Content Mining describes the discovery of useful information from the Web contents/data/documents.

Web structure mining tries to discover the model underlying the link structure on the Web.

Finally, Web usage mining tries to make sense of the data generated by the Web surfer's sessions or ⊆behaviors. While the Web content and structure mining utilize the real or primary data on the Web, Web usage mining mines the secondary data derived from the interactions of the users while interacting with the Web.

2.3. Background and Terminology on FCA

In FCA, a triple *(O,M, I)* is called a context, where $O=\{g_1, g_2, \ldots, g_n\}$ is a set of n elements, called objects; $M = \{1, 2, \ldots, m\}$ is a set of m elements, called attributes; and $I \subseteq O \times M$ is a binary relation. The context is often represented by a cross-table as shown in Fig 1 [13].

A set $X \subseteq O$ is called an object set, and a set $J \subseteq M$ is called an attribute set. Following the convention, we write an object set *{a, c, e}* as *ace*, and an attribute set *{1,3,4}* as *134*. For $I \in M$, denote the adjacency list of i by $nbr(i)=g \in O:(g,i) \in I\}$.Similarly, for $g \in O$, denote the adjacency list of g by $nbr(g)=\{I \in M:(g,i) \in I\}$.

The function attr : $2^O \rightarrow 2^M$ maps a set of objects to their common attributes: $attr(X) = \cap_{g \in X} nbr(g)$, for X $\subseteq$ O. The function obj : $2^M \rightarrow 2^O$ maps a set of attributes to their common objects: $obj(J) = \cap_{j \in J} nbr(j)$, for $J \subseteq$

M. It is easy to check that for $X \subseteq O$, $X \subseteq obj(attr(X))$, and for $J \subseteq M$, $J \subseteq attr(obj(J))$. An object set $X \subseteq O$ is closed if $X = obj(attr(X))$. An attribute set $J \subseteq M$ is closed if $J = attr(obj(J))$. The composition of *obj* and *attr* induces a Galois connection between 2^O and 2^M.

Readers are referred to [13] for properties of the Galois connection.

A pair $C = (A,B)$, with $A \subseteq O$ and $B \subseteq M$, is called a concept if $A = attr(B)$ and $B = obj(A)$.

For a concept $C = (A,B)$, by definition, both A and B are closed. The object set A is called the extent of C, written as $A = ext(C)$, and the attribute set B is called the intent of C, and written as $B = int(C)$. The set of all concepts of the context (O,M, I) is denoted by $B(O,M, I)$ or simply B when the context is understood.

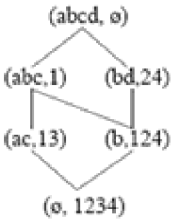
Let (A_1,B_1) and (A_2,B_2) be two concepts in B. Observe that if $A_1 \subseteq A_2$, then $B_2 \subseteq B_1$. We order the concepts in B by the following relation $<$:

$$(A_1,B_1) < (A_2,B_2) \leftrightarrow A1 \subseteq A2 (B_2 \subseteq B_1).$$

It is not difficult to see that the relation $<$ is a partial order on B. In fact, $L = < B, <>$ is a complete lattice and it is known as the concept or Galois lattice of the context (O,M, I). For $C,D \in B$ with $C < D$, if for all $E \in B$ such that $C < E < D$ implies that $E = C$ *or* $E = D$, then C is called the *successor* (or lower neighbor) of D, and D is called the *predecessor* (or upper neighbor) of C . The diagram representing an ordered set (where only successors/predecessors are connected by edges) is called a Hasse diagram (or a line diagram). See Figure 1 for an example of the line diagram of a Galois lattice.

	1	2	3	4
a	×		×	
b	×	×		×
c	×		×	
d		×		×

(a) A context (O,M, I) with O = {a, b, c, d} and M={1,2,3,4}. The cross × indicates a pair in the relation I.



(b) The corresponding Galois/concept lattice

Figure 1. A formal context and the corresponding concept lattice

For a concept $C = (ext(C), int(C))$, $ext(C) = obj(int(C))$ and $int(C) = attr(ext(C))$. Thus, C is uniquely determined by either its extent, $ext(C)$, or by its intent, $int(C)$.We denote the concepts restricted to the objects O by $B_O = \{ext(C): C \in B\}$, and the attributes M by $B_M = \{int(C): C \in B\}$. For $A \in B_O$, the corresponding concept is $(A, attr(A))$. For $J \in B_M$, the corresponding concept is $(obj(J), J)$. The order $<$ is completely determined by the inclusion order on 2^O or equivalently by the reverse inclusion order on 2^M. That is, $L = <B,<>$ and $L_M = < B_M, \supseteq >$ are order-isomorphic.

We have the property that $(obj(Z), Z)$ is a successor of $(obj(X), X)$ in L if and only if Z is a successor of X in L_M. Since the set of all concepts is finite, the lattice order relation is completely determined by the covering (successor/predecessor) relation. Thus, to construct the lattice, it is sufficient to compute all concepts and identify all successors of each concept.

3. Limitless of Web Mining

When interacting with the Web, the following problems are presented:

- Finding relevant information, so, people either browse or use the search service when they want to find specific information on the Web. When a user uses search service, called search engine, he usually inputs a simple keyword query and the query response is the list of pages ranked based on their similarity to the query. However, proposed search tools have the following limits. We mention: i) the low precision which is due to the irrelevance of many of the search results. This results in a difficulty finding the relevant information, ii) the low recall which is due to the inability to index all the information available on the Web. This results in a difficulty finding the unindexed information that is relevant.
- Creating new knowledge out of the information available on the Web: Actually this problem could be regarded as a sub-problem of the problem above. While the problem above is usually a query-triggered process (retrieval oriented), this problem is a data-triggered process that presumes that we already have a collection of Web data and we want to extract potentially useful knowledge out of it (data mining oriented). Recent research focuses on utilizing the Web as a knowledge base for decision making.
- Personalization of the information: this problem is often associated with the type and presentation of information, since it is likely that people differ in the contents and presentations they prefer while interacting with the Web. On the other hand, the information providers could encounter these problems, among others, when trying to achieve their goals on the Web.

In spite of the great success which knew FCA in the field of Data mining, rare are the search engines which use this theory in their indexing step.

4. A New IR Model

In this section, an IR model will be presented. It consists in the application of FCA method for relevant information search from WWW. For accessing at relevant information, we followed a cycle described in the figure 2.

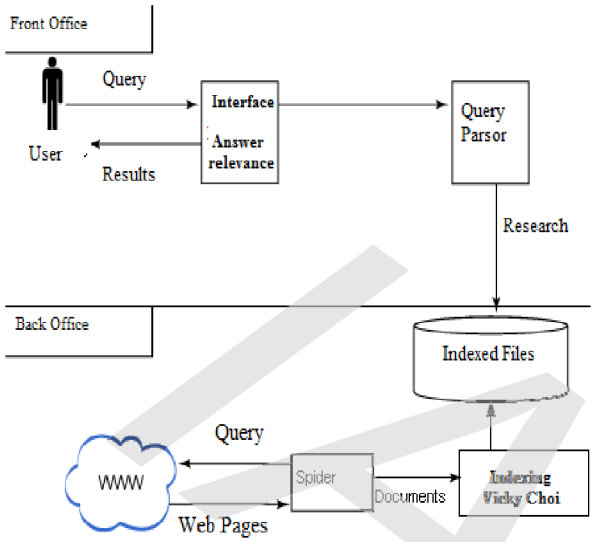


Figure 2. Information retrieval cycle on the Web.

We have adopted, like Web Mining category, Web content mining. Data type supported is textual or multimedia (HTML, Pdf, Doc, Png, Mp3... etc.),
Our information retrieval model is made up mainly of four parts in conformity with the KDD process: the crawler, the indexer, the searcher and the display.

4.1. The Crawler

Crawlers are robots which traverse Web sites while following the links of the million Web pages. The crawler assigns a priority to each Web page according to the links of which it is made up. These links are saved in a file whose this last will be defines as entered of the indexing step.

In this step, the techniques of data cleaning are applied. Indeed, it acts to clean the links which will not correspond to our search needs.

The links to be removed are the doubled blooms (links towards the same target in the same Web page, the links towards private sites....).

4.2. The Indexer

It is a very determining step for the information search. Indeed, if we need to cover a broad field of textual file or to find a character string precise in only one file, scanner is not needed sequentially each file for the given sentence. Because the file number is large, it is better to establish an index of the texts in a format which allows fast search, which avoids the sequential method.

This step is parallel at the crawler step, indeed, to index new resources, a robot proceeds by following the founded hyperlinks starting from a pivot page. Thereafter, it is advantageous to memorize the URL of each recovered resource and to adapt the visit's frequency to the observed frequency of resource's update.

An exclusion file placed in the root of a Web site makes it possible to give to the robots a list of resources to be ignored (cleaning). This convention makes it possible to reduce the Web server charge and to avoid resources without interest.

This step is ensured through an algorithm such as Vicky Choi [2], while respecting the FCA method. Indices are expressions formed by the titles and textual contents of the Web pages. They have the tree structure.

Indeed, it is a question of building a hierarchy which defines the Web navigation structure. This tree corresponds to the concepts lattice construction starting from a context.

Within the meaning of concepts lattice, core of FCA, and in our particular context, a formal concept will be a set of words and a set of documents in which all the words are in all the documents and, conversely, in which all the documents contain all the words.

Example: Let us take the example of 10 words or terms and 6 documents. Let {a,b,c,d,e} the set of the documents and {1,2...,7} the set of the words. We will test the membership of the latter to each document in order to generate the formal context (see table 1).

Table 1. Formal context.

	1	2	3	4	5	6	7
a	X					X	
b	X		X	X	X	X	
c	X			X		X	
d		X	X		X		
e		X					X

Recall that constructing a concept lattice includes generating all concepts and identifying each concept's successors.

The algorithm starts with the top concept *(O,attr(O))*. It process the concept by computing all its successors, and then recursively process each successor by either the Depth First Search (DFS) order - the ordering obtained by DFS traversal of the lattice - or Breadth First Search (BFS) order. Successors of a concept are computed from its children.

Let $C = (obj(X),X)$ be a concept. First, it compute all the children $Child(C) = \{(obj(XS),XS) : S \in AttrChild(X)\}$. Then for each $S \in AttrChild(X)$, it checks if XS is closed. If XS is closed, $(obj(XS),XS)$ is a successor of C.

Since a concept can have several predecessors, it can be generated several times. It checks its existence to make sure that each concept is processed once and

only once. The pseudo-code of the algorithm based on BFS is shown in Algorithm 1 [2].

Algorithm 1 CONCEPT-LATTICE CONSTRUCTION – BFS

1: Compute the top concept $C = (O, attr(O))$;
2: Initialize a queue $Q = \{C\}$;
3: Compute Child(C);
4: **while** Q is not empty **do**
5: C = dequeue(Q);
Let X = int(C) and suppose AttrChild(X) =< $S1$, $S2$, , Sk >;
6: **for** i = 1 to k **do**
7: **if** XSi is closed **then**
Denote the concept (obj(XSi),XSi) by K;
8: **if** K does not exist **then**
9: Compute Child(K);
10: Enqueue K to Q;
11: **end if**
12: Identify K as a successor of C;
13: **end if**
14: **end for**
15: **end while**

Generated concepts lattice is represented in Figure 3.

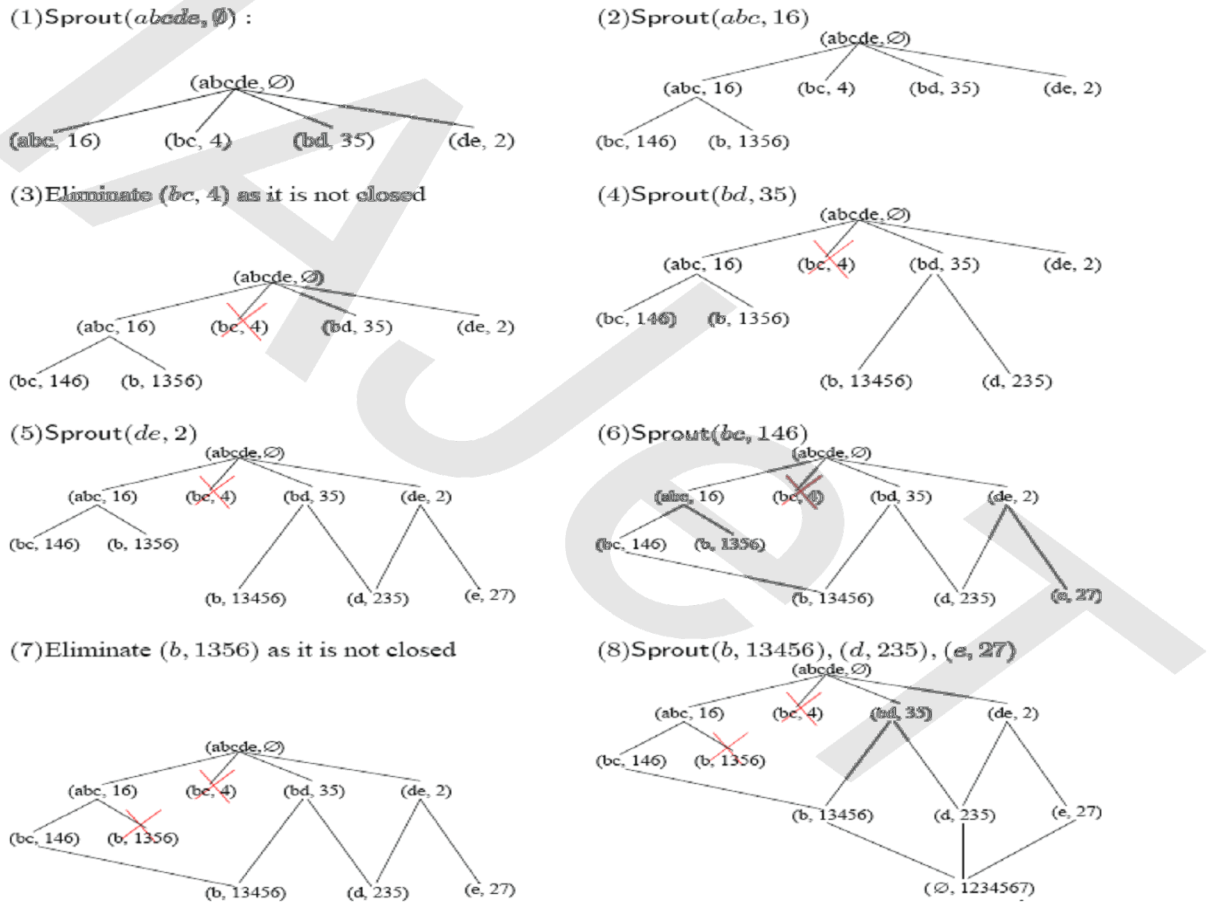


Figure 3. Execution of Vicky Choi algorithm.

4.3. The Searcher

The searcher receives a query in the form of list of words or terms supplied with a priority.

When the user of a search engine fills the form, it specifies the words which it seeks (possibly those which it does not wish) thanks to the Boolean operators "AND", "OR", "NOT"... (Symbolized by +, -,...).

Research is the action to look at words in an index to find references to documents when they appear.

The quality of a research is evaluated by the positioning and the relevance of the results. However, other factors enter in account a search. The speed is a factor to treat a huge quantity of information.

In the same way, capacity to support simple or complex queries, sentences querying, characters, the results of positioning and sorting are as significant as an easy syntax to take in hand to enter these queries.

The rules which will follow us to form a regular expression are:

- If there is several operators between two successive words, only the first will be kept,
- If there is not operator between two successive words, operator "AND" is inserted,

- Automatic deletion of words not having values in research.

Let us take an example of research for information following the execution of a query. Let the documents A, B, C... etc, containing the words "Security", "Internet", "Network" and "Company". Then sets (Security, Internet, Networks, Company) and (A, B, C, D, E, F, G, H) form a formal concept.

A concepts lattice is built starting from all the found concepts and it constitutes a representation of the agreement between the documents and the words. Example: Let the context table presented by Table 2:

Table 2. Example of Context related to a Query.

		Terms			
		Security	Internet	Network	Company
DOCUMENTS	A	0	1	1	0
	B	1	0	0	0
	C	0	0	1	1
	D	1	0	0	0
	E	0	0	0	1
	F	1	1	1	1
	G	0	1	1	0
	H	0	0	1	0

Figure 4 illustrates an example of lattice generated starting from a query "Security Internet of the networks in the company".

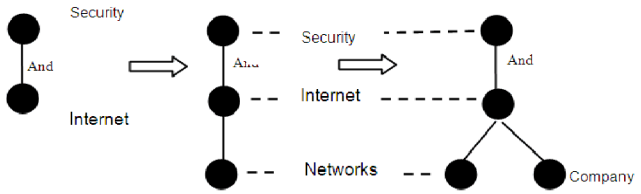


Figure 4. Example of concepts lattice construction.

Then we build Hyper Index which will be the union of the following lattices: (see Figure 5).

4.4. The Displayer

The display is a HTTP server. It receives query on behalf of the customers subjecting queries to him, and returns HTML pages containing the results of research. This page contains links making it possible to the user to go directly on the results pages or to consult the pages out of mask.

5. Implementation and Tests

5.1. Search Engine principle use

Figure 65 expresses the principle of use of our model.

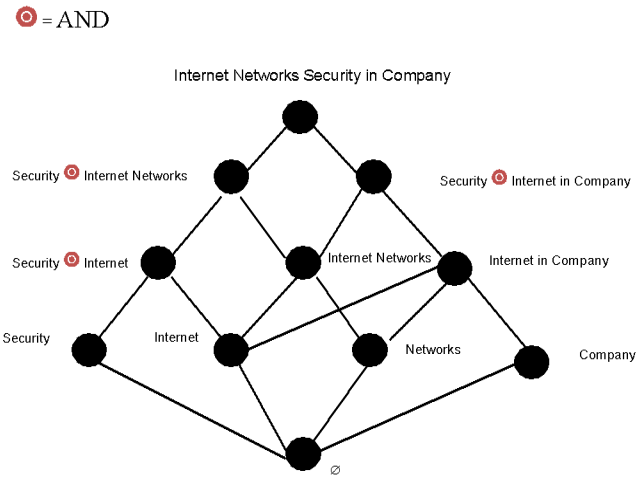


Figure 5. Hyper Index Construction

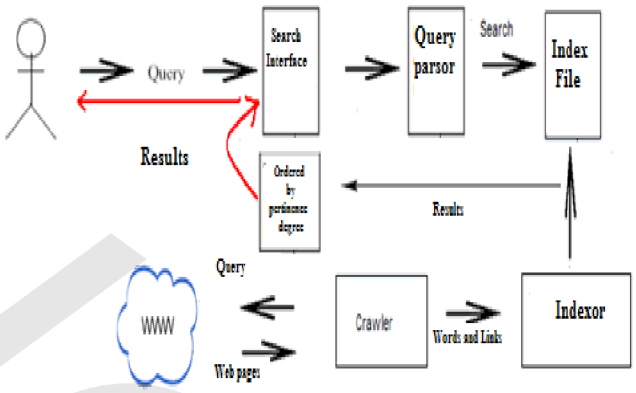


Figure 6. Principle use of the proposed Search Engine.

Following the process of Web site links extraction and indexing of the contents of each one of them, the user launches a search through the input data of a traditional query on the search interface, a query parsor analyzes this and determines the combination of the words to seek, then, a search is made in the index file and returns results to user, the sorting of these results is necessary for a better display of the data and this through relevance index calculation of each result and display of the results according to the following rules:

- The place of the formulated words in the query,
- Frequency of the scanned words,
- Contents of the URL,
- The key word order in the documents,
- The size of the document,
- The modification/creation date of the documents.

5.2. Model Architecture

Our search model is composed of four principal parts: the browser (or crawler), the indexer, the searcher and the display. These parts are presented in figure 7.

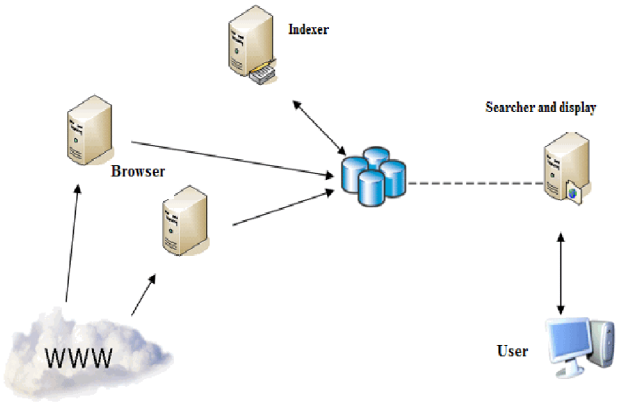


Figure 7. Operating mode of the Web Search Model.

Each part is independent. It is thus possible to make turn the programs into simultaneous or ones after the others.

The crawler is charged to traverse the Web server by introducing a start address from which it will download the links contained in the Web pages, and for each found link, it remakes the same thing until reaching level *N* of Web pages according to the navigation structure defined in those.

It is about the collection of the resources (Web pages, Pdf document, Txt, Doc... etc) in order to make it possible the search engine to index them. With each visited page, the crawler assigns a priority to this one according to the words and links of which it is made up.

This step will not be carried out with each search (i.e. at each user query), but rather according to the strategy of update defined by the Webmaster.

Indeed, this step is dependent in execution time on the band-width of the network as of existing large data on the Web server.

We associates to this step a data cleaning through the installation of a an exclusion file (robots.txt) placed in the root of a Web site allowing to give to the crawler a list of resources to be ignored. This convention makes it possible to reduce the load of the Web server and to avoid resources without interest. A very great number of pages are added, modified and removed each day. If the storage capacity of information, like the speed of the processors, increased quickly, the band-width did not profit from the same progression. The problem is thus to treat a volume always crescent of information with a limited flow. The robot needs to give priorities to its downloading.

Continuation to the Web crawling step, an indexing step is required. Indeed, the indexing of the crawled pages consists in extracting the words from them. We obtain a list of all words contained in the crawled pages. For each word, we associate the list of pages which contain this word and we specify the number of appearance of this word in the concerned pages. This number will play a significant role in the display of the results of a query user, and it is at this step that the part of the searcher intervenes. Indeed, the latter receives a query in the form of a list of terms supplied with a priority. The terms are charged starting from the dictionary (the index already built), then we calculate the row (index of relevance or score) of the documents containing the given terms and we display the results sorted by row on a Web page, and this is ensured by the display part. The latter is a HTTP server which receives queries on behalf of the subjecting customers' queries to him, and returns HTML pages containing the search results. This page contains links making it possible to the user to go directly on the results pages.

5.3. Operating Mode

The graphic interface is the most significant part in the realization of our model offering to the user an easy search interface to be handled.

The search engine administrator is responsible for the index database update through an interface dedicated to this.

Figure 8 describes the parameters setting interface in the proposed search model.

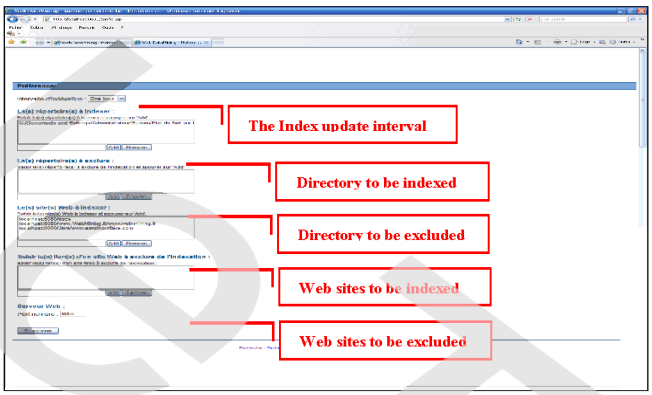


Figure 8. Parameter's Setting Interface.

Through this interface, the administrator can seize the list of the repertories and Web sites to be indexed. He can also define as a preliminary the repertories and links to be excluded from the indexing (cleaning). All these parameters will be recorded in an exclusion "Robots.txt" file.

During the recording, the crawling and indexing steps will be started. The follow-ups of the course statute of these steps are controlled through an interface which is updated every five seconds. It is illustrated in Figure 9.

Once the finished indexing, the index is present on server. Users can carry out a research through the entry words or expressions. Figure 10 shows this research interface.

The research results will be displayed in an interface of which this one well represents from top to bottom the classified results by their score. A time response of the server is calculated to show the relevance of our model which is represented on the result page as well as the number of found results: (See Figure 11)

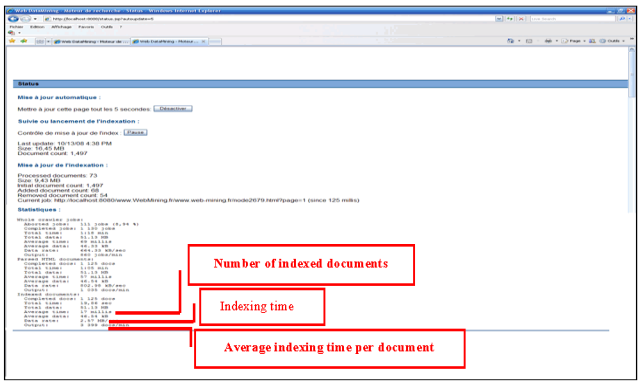


Figure 9. Crawling and Indexing follow-up Interface.

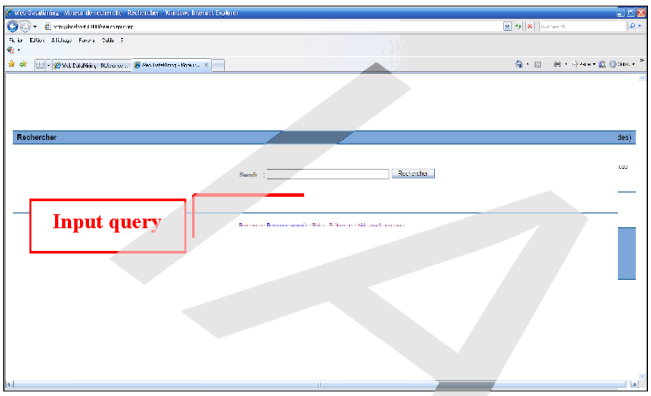


Figure 10. User Query Interface.

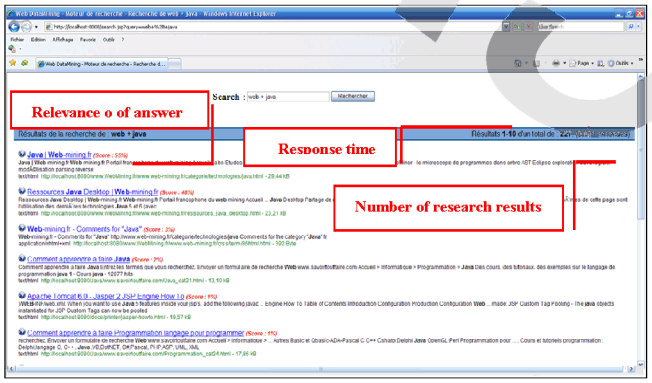


Figure 11. Search results.

An advanced research mode is possible through our model. Indeed, the user can filter the results of its research by specifying the type of documents to be displayed.

4. Evaluation

The proposed search model made it possible to proceed in the relevant information in order to improve offered services by the Web sites.

Performances of the algorithms were related to the following factors:

- The crescent volume of information,
- The network flow,
- The storage capacity of information,
- The speed of the processors.
- The assessment showed that:
- IR using FCA is as easy as a sequential and Boolean search,
- The data organization is as simple as the traditional systems (to associate the new objects of the attributes, to add and cut off from the objects),
- The server response is minimal (about seconds) and this reflects the good choice of the indexing used Vicky Choi algorithm [2],
- Conceptual classification of the search results while allowing the user the comparison between displayed results thanks to the relevance degree displayed showing the importance of each result.

In order to test the performance of our model, a comparative analysis of research functions between some research engines is carried out and presented in table 3.

5. Conclusion

The limited capacity storage, the Web data knowledge extraction problem, the need for relevant information search in a Web site, are as many factors which require the recourse to advanced search techniques.

The majority of the marketed search engines use algorithms of more or less complex indexing for the IR.

In spite of the great success which knew the FCA method based on lattice theory, in the field of Data Mining, rare are the search models which use this concept in their indexing step.

For this, we have proposed a new search model which consists in working out a step based on the FCA method. We manage to obtain a conceptual classification being pressed on the Vicky Choi [2] algorithm and the concepts lattice which it generates.

Indeed, we have informed on the combined use of this last with the Web Mining methods and this enabled us to reflect the importance of the indexing of the Web data for information retrieval.

Implementation was complex and its comparison with other search model was difficult sight that this latter do not clear up on the size and sources of their index databases.

However, several improvements remain to make. Indeed, the representation of the concepts lattice becomes very complex if the number of concepts exceeds the hundreds. One of the possible solutions to ensure a good data presentation would be to carry out the application conceptual scaling method. With this solution, each node of the lattice would represent itself then a lattice.

Table 3. Comparison Research Functions between some Search Engines.

Research engine	Required contents	Advanced research Modulate	Multilingual	Boolean operators	Truncation	Sensitive to breakage (case sensitive)	Exact expression	Displayed relevance degree	Truncation
ALL THE WEB	Full text Web pages, pdf files, Word files, flash	Yes	No	AND (by default) - (removes word that does not want) + (compulsory the word makes which follows)	No	No	" "	No	No
ALTAVISTA	Full text Web pages	Yes	Yes	AND (by default) - (removes word that does not want) + (compulsory the word makes which follows) AND, OR, AND NOT, NEAR, Nesting (in the advanced research mode) Possibility to use natural language	*	Yes	" "	No	*
TEOMA	Full text Web pages	Yes	No	AND (by default) OR (en majuscules) - (removes word that does not want) + (compulsory the word makes which follows) " " (to locate blank words)	No	No	" "	No	No
WISENUT	Full text Web pages	No	No	- (removes word that does not want) + (compulsory the word makes which follows)	No	No	" "	No	No
Our model	Full text Web pages, PDF files, Microsoft Office documents (WORD, EXCEL, POWER POINT), PostScript and others files. Research in the the hard disk epertories	Yes	No	AND (by default) and OR - (removes word that does not want) + (compulsory the word makes which follows)	*	No	" "	Yes	*

References

[1] Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Kumar, S., Raghavan, P., Rajagopalan, S. and Tomkins, A. (1999), "Mining the link structure of the World Wide Web" *IEEE Computer*, Vol. 32(8), pp. 60-67.

[2] Choi, V., Zheng, C., Zhu, Q., Sankoff, D. (2007) "Algorithms for the Extraction of Synteny Blocks from Comparative Maps", *WABI*, pp. 277-288.

[3] Etzioni, O. (1999), "The World Wide Web: quagmire or gold mine" *Communications of the ACM*, Vol 39(11).

[4] Liu, B. (2006) "Web Data Mining, Exploring Hyperlinks" *Contents and Usage Data*, Springer.

[5] Madria, S.K., Bhowmick, S.S., Ng, W.K. and Lim, E.-P. (1999) "Research issues in Web Data Mining", *Proceeding of Data Warehousing and Knowledge Discovery*, pp. 303-312.

[6] Messai, N. (2004) "Treillis de Galois et ontologies de domaine pour la classification et la recherche de sources de données génomiques", *DEA informatique de Lorraine-Ecole Doctorale IAEM Lorraine*.

[7] Nguifo, M. (2005) "Treillis de concepts et classification supervisée : un état de l'art", CRIL rapport de recherche.

[8] Omarjee, S. (2002) "Le Data Mining" *Aspects juridiques de l'intelligence artificielle au regard de la protection des données personnelles*, Université Montpelier.

[9] Spiliopoulou, M. (1999) "Data mining for the Web", *Principles of Data Mining and Knowledge Discovery*, Second European Symposium, pp. 588-589.

[10] Srivastava, J., Cooley, R., Deshpande, M. and Tan, P.-N. (2000) "Web usage mining: Discovery and applications of usage patterns from Web data", *SIGKDD Explorations*, Vol. 1(2).

[11] Uthurusamy, R. (1996) "From Data Mining to Knowledge Discovery" Current challenges and future directions, *In advances in Knowledge Discovery and Data Mining*, pp. 561-569.

[12] Vaithyanathan, S. (1999) "Introduction: Data Mining on the Internet", *Artificial Intelligence Review*, Vol. 13(5/6), pp. 343-344.

[13] Wille, R. (1999) "Formal Concept Analysis", *Mathématical fondations*, Springer Verlag.

Minyar Sassi Hidri received the diploma of engineering in computer science and Ph.D. in electric genius from the National Engineering School of Tunis, Tunisia in 2003 and 2007, respectively. Actually, Dr. M. Sassi Hidri is an assistant professor at the Department of Technologies of Information and Communications in the National Engineering School of Tunis. She is also a member of the Systems and Signal Processing Laboratory (LSTS). Her researches interest includes many aspects of query optimisation, Data Mining methods and database flexible querying.

Amel Grissa Touzi received her PhD and HDR in Computer Science from the Faculty of Sciences of Tunis, Tunisia in 1994 and 2010, respectively. She is a Professor at the Department of Technologies of Information and Communications in the National School of Engineering of Tunis, Tunisia. She is also a member of the Systems and Signal Processing Laboratory (LSTS). Her research interest includes many aspects of deductive databases, fuzzy databases, and flexible querying.