# Structured Conceptual Meta-Search Engine

Ali Jaoua[1], M. Lazhar Saidi[1], Ahmed Hasnah[1], Jihad Mohamad AlJa'am[1], Sahar Ahmed[1], Baina Salem[1], Noura Rashid[1], Shereen Shareef[1], Suad Zaghlan[1]

[1] Computer Science and Engineering Department,
College of Engineering, Qatar University
{jaoua, yazid, hasnah, jaam}@qu.edu.qa

**Abstract.** This paper presents a Web-based Internet meta-search engine application based on Formal Concept Analysis. The implemented prototype combines the advantages of Web searching and Formal Concept Analysis (FCA). Receiving at real time the list of URLs as results of Google and Yahoo search engines. The developed meta-search engine makes a conceptual clustering of these results, and exploits the ranking given by Google and Yahoo. Then, it generates a tree of "optimal" concepts through which the users can browse easily the results to converge as fast as possible to their needs. A meta-search engine has been experimented by many potential users and has given good satisfaction. It is now considered suitable for a lot of improvements and as a prototype for intensive research on web engineering, and information retrieval.

**Keywords:** Concept Analysis, Meta Search Engine, Clustering.

## 1 Introduction

While browsing the Internet, the most important need for the user is to find pertinent information in the shortest possible time. Generally, extracting pertinent information from data requires mainly the two following tasks: first read and classify data, second select the most suitable information related to the user interest. Computers and communication systems are mainly used to search and retrieve URLs with very high speed from allover the world, creating obviously the need for developing a layer of information engineering software (i.e. "intelligent software") which main task is to read and organize data for the user, at real time. These intelligent systems have the precious task to classify dynamically and incrementally new arriving URLs or data. They are dedicated to make repetitive classification activities, preparing the work to the human browser, and presenting it with a more understandable and structured view.

While conceptual clustering [2,3,5] is a good approach for URLs classification with respect to the semantic level: (i.e. the meaning of words), ranking offered by most of search engine might be used to assess the importance or priority of a document. As a matter of fact, several similar meta search engines have been realized using different or similar conceptual methods as Credo [2] where authors use the complete Galois Lattice of concepts. In our system, we preferred to check if selecting

only “optimal” concepts" gives the most pertinent URL's. We also think that only given first optimal concepts are significant, and therefore users might save time during the browsing step. Here, we do not try to build all the Galois lattice of concepts to make a structured view, but we only extract a subset of concepts which might cover all the initial binary concept, then we build a hierarchy based on the rank of each concept, dynamically calculated. The next section includes formal concept analysis and relational algebra foundations, the mathematical foundations used in this work. In the third section, using an example, we give an approximate algorithm to extract optimal concepts. We also explain how we rank concepts, URLs and keywords, and how we assign a title (or name) to the selected concepts. In section 4, we give an outline of the system structure. In section 5, we explain how the implemented meta-search engine is used in practice and our first impression about its performance. Finally we conclude and suggest some improvement of the proposed system in the future.

## 2 Background and Mathematical Foundation

### 2.1 Formal Concept Analysis

The Formal Concept Analysis (*FCA*) is a theory of data analysis which identifies conceptual structures among data sets. It was introduced by Rudolf Wille [1,5] and has since then grown rapidly.

Let $G$ be a set of objects and $M$ be a set of properties. Let $I$ be a binary relation defined on the set $E$. For two sets $A$ and $B$ such that $A \subseteq E$ and $B \subseteq E$, we defined two operators $f(A) = A^R$ and $h(B) = B^Q$ as follow:

| $f(A) = A^R = \{m \mid \forall\, g \in A \Rightarrow (g, m) \in I\}$ | $h(B)=B^Q = \{g \mid \forall m \in B \Rightarrow (g, m) \in I\}$ |
|---|---|

A formal context $k := (G,M,I)$ consists of two sets $G$ (objects) and $M$ (Attributes) and a relation $I$ between $G$ and $M$. Formal Concept of the context $(G,M,I)$ is a pair $(A,B)$ with: $A \subseteq G$, $B \subseteq M$, $A^R = B$ and $B^Q = A$. We call A the extent and B the intent of the concept $(A, B)$. IF $(A_1, B_1)$ and $(A_2, B_2)$ are two concepts of a context, $(A_1, B_1)$ is called a sub concept of $(A_2, B_2)$, provided that $A_1 \subseteq A_2$ and $B_2 \subseteq B_1$. In this case, $(A_2, B_2)$ is a super concept $(A_1, B_1)$ and it is written $(A_1, B_1) < (A_2, B_2)$. The relation “<” is called the hierarchical order of the concepts .The set of all concepts of $(G, M, I)$ ordered in this way is called the concept lattice of the Context $(G, M, I)$.

### 2.2 Relational Algebra

A binary relation $R$ between two finite sets $D$ and $T$ is a subset of the Cartesian product $D \times T$. An element in $R$ is denoted by $(x,y)$, where $x$ designate the antecedent and $y$ the image of $x$ by $R$.

For a binary relation we associate the subsets given as follows: The set of images of $e$ defined by: $e.R = \{e' \mid (e, e') \in R\}$. The set of antecedents of $e'$ defined by: $R.e' = \{e \mid (e, e') \in R\}$; The domain of $R$ is defined by: $Dom(R) = \{e \mid (e, e') \in R\}$. The range of $R$ is defined by: $Cod(R) = \{e' \mid (e, e') \in R\}\}$; The cardinally of $R$ defined by: $Card(R) = \{\text{numbers of pairs in } R\}$. Let $R$ and $R'$ be two binary relations, we define

the relative product (or setting up) of $R$ and $R'$, the relation $R$ o $R' = \{(e, e') \mid$ It exists $t$ in cod($R$) $\mid (e, t) \in R$ & $(t, e') \in R'\}$, where the symbol "o" represents the relative product operator. The inverse relation of $R$ is given by: $R^{-1} = \{(e, e') \mid (e', e) \in R\}$. The relation $I$, identity of a set $A$ is given: $I(A) = \{(e, e) \mid e \in A\}$. The complement of the binary relation $R$ is given by the following: $\{(e, e') / (e, e')$ is an element of $R\}$.

### 2.3 Notion of Rectangle

Let $R$ be a binary relation defined between $D$ and $T$: A rectangle of $R$ is a Cartesian product of two sets $(A,B)$ such that $A \subseteq D$, $B \subseteq T$ and $A$x$B \subseteq R$, $A$ is the domain of the rectangle and $B$ is the range. The rectangle closure $R^*$ of a binary relation is defined by the product of the domain and range of $R$ (Cod(R)): $R^* = \text{Dom}(R)$ x $\text{Cod}(R)$.

**Definition 1.** Let $R$ be a binary relation defined between $D$ and $T$. A rectangle $(A,B)$ of $R$ is *maximal* if wherever $A$ x $B \subseteq A'$x$B' \subseteq R$, we have $A = A'$ and $B = B'$.

### 2.4 Elementary Relations

If $R$ is a finite binary relation (i.e., subset of $E$ x $F$, where $E$ is a set of objects and $F$ a set of properties) and $(a, b) \in R$, then the union of rectangles containing (a, b) is the elementary relation $PR$ (i.e. subset of $R$) given by:

$$PR = \Phi_R(a,b) = I(b.R^{-1}) \text{ o } R \text{ o } I(a.R)$$

Where $I$ is the identity relation**,** $R^{-1}$ is the inverse relation of $R$ (i.e. set of inversed pairs of $R$), and "o" refers to the relative product. $PR$ is the sub-relation of $R$, pre-restricted by the antecedents of $b$ (i.e. $b.R^{-1}$), and post-restricted by the set of images of a (i.e. $a.R$).

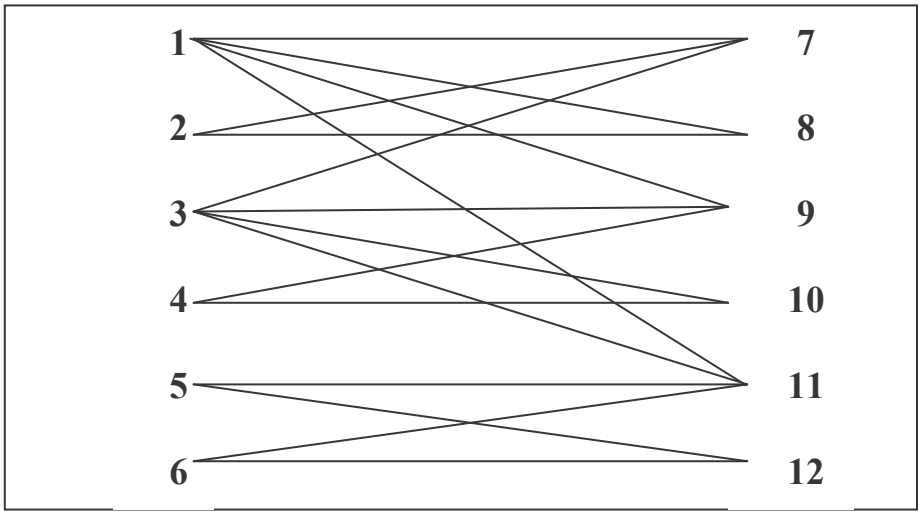
**Definition 2 (Gain in Storage space)**. The gain in storage space $W(R)$ of binary relation is given by: $W(R) = (r/dc)\ (r\text{-}(d+c))$, where, $r$ is the cardinality of $R$ (i.e. the number of pairs in binary relation $R$), $d$ is the cardinality of the domain of $R$, and $c$ is the cardinality of the range of $R$.

*Remark*. The quantity $(r/dc)$ provides a measure of the density of the relation $R$. The quantity $(r\text{-}(d+c))$ is a measure of the economy of information.

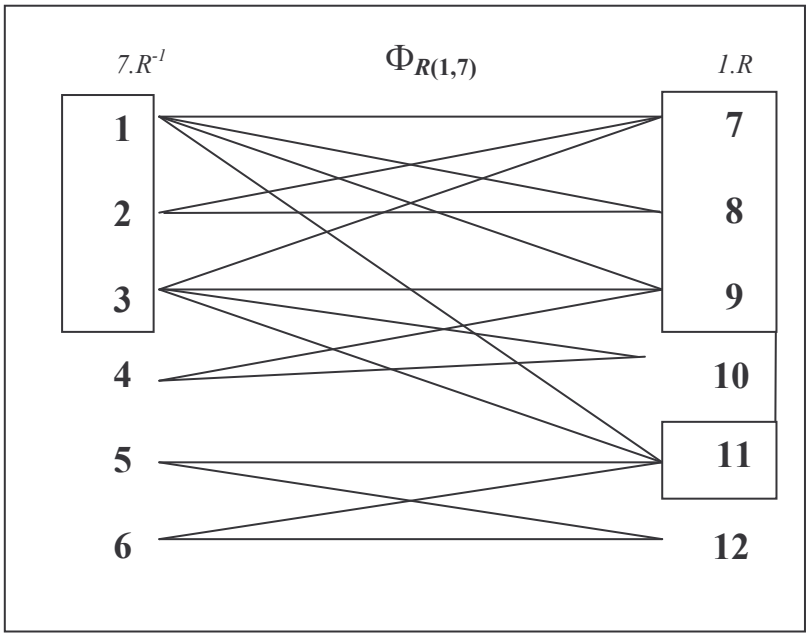
**Definition 3 (Optimal Concepts)**. A maximal rectangle $RE \subseteq R$ containing an element $(x,y)$ of a relation $R$ is called optimal if it produces a maximal gain $W(RE(x,y))$ with respect to other concepts containing $(x,y)$. As a matter of fact a concept RE embedded in a binary relation has as cardinality $d$ x $c$, (i.e, needs $d$ x $c$ links to be represented), when the concept is extracted it might be represented by only one link between its domains (i.e $d$ elements) and its range (i.e. $c$ elements), and therefore would need a space of $(d+c)$ elements. In this case: $W(RE) = d$ x $c - (d + c)$.

## 3 An Efficient Algorithm for Optimal Concept Extraction

In this section, we only explain an algorithm for optimal concept extraction applied to an example. Let $R$ be a finite binary relation between two sets relating 6 URLs to 12 Keywords, as illustrated below:




Assume that we want to select an optimal concept containing the pair (1,7) in $R$:



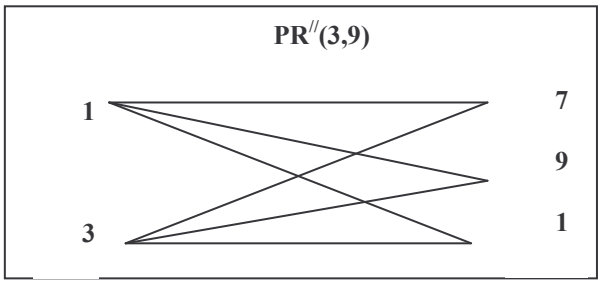
The elementary relation of (1,7) is the following: $PR(1,7) = \Phi_{R(1,7)} = I\,(7.R^{-1})$ o $R$ o $I$ (1.$R$). So we search with an iterative way the optimal rectangles of $PR$ (1,7) which successively contains the elements (1,8), (1,9), (1,11), (2,7) and (3,7).

**First Iteration**: from the five elementary relations of the above mentioned elements select the first that gives a maximal gain:
1. $PR'_{1,8} = \Phi_{PR1,7}\,(1,8)$; $W(PR'_{1,8}) = 0$
2. $PR'_{1,9} = \Phi_{PR1,7}\,(1,9)$; $W(PR'_{1,9}) = 7/8 \rightarrow$ **Selected**
3. $PR'_{1,11} = \Phi_{PR1,7}\,(1,11)$; $W(PR'_{1,11}) = 7/8$
4. $PR'_{2,7} = \Phi_{PR1,7}\,(2,7)$; $W(PR'_{2,7}) = 0$
5. $PR'_{3,7} = \Phi_{PR1,7}\,(3,7)$; $W(PR'_{3,7}) = 7/9$

The selected elementary relation $PR'_{1,9}$ is not a rectangle, so the algorithm continues on the previous selected elements namely (1,7) and (1,9).

**Second Iteration:** Search now the optimal rectangles of $PR'_{1,9}$ that successively contain the elements (1,8), (1,11) and (3,9). This step provides three elementary relations:
1. $PR''_{1,8} = \Phi_{PR'_{1,9}}\,(1,8)$; $W\,(PR''_{1,8}) = -1$
2. $PR''_{1,11} = \Phi_{PR'_{1,9}}\,(1,11)$; $W\,(PR''_{1,11}) = 7/8$
3. $PR''_{3,9} = \Phi_{PR'_{1,9}}\,(3,9)$; $W(PR''_{3,9}) = 1$
   **→ selected.**

$PR''_{3,9}$ is a rectangle, so it is optimal and contains the element (1,7) of $R$.




### 3.1 URL and Optimal Concept naming and ranking

Ranking of search results is a vital parameter that reflects how relevant these results might be to the user. In our approach, the most optimal concept is presented first to the user. The rank or order of a concept $RE$ is defined as its gain $W(RE)$. We think that the choice of the degree of optimality or gain of a concept (defined in the previous section) as a ranking measure better serves the user interest.. The rank of URLs is calculated as a function of the original ranks supplied by the provider search engines and the total size of results found by them, as in the following formula:

$$Rank = [Size1*Rank1 + Size2 * Rank2] / [Size1+Size2]$$

Where *Rank1* (respectively *Rank2*) is the rank of the URL and *Size1* (respectively *Size2*) is the total number of URLs as given by the first (respectively the second) search engine. This formula is an attempt to eliminate dependency of the URLs ranks on the size of results returned from the search engines. Keywords also need to be ranked so that the one with the highest rank indexing some URL is selected to represent a URL or a Concept and accordingly search results can be more readable and meaningful. The Keyword rank is calculated after the process of ranking URLs. The rank of each keyword is defined as the highest URL's rank in the set of URLs associated with this keyword. Accordingly, each URL is named by the keyword that has the highest rank. The same process is repeated for each concept. The concept name is the highest rank keyword associated with the concept.

## 4 Meta-Search Engine System Structure

The system enables the user to enter a query, which is passed to the integrated web service API provided by two search engines: Google and Yahoo. As a result, two lists of URL's with their description are returned from the APIs. The system processes these URL's with their description, first by tokenizing each description string into several keywords after eliminating the empty words and redundancy. Then a binary relation (*BR*) is created as a hash table containing keywords and a list of URLs with associated keywords. From binary relation BR, the most important optimal concepts are extracted using the algorithm presented in section 3, where an optimal concept is defined as the most economical full association of a subset of URLs with a subset of keywords. Exploiting concepts, we derive a structured space where users may browse dynamically to converge as fast as possible to their needs. From binary relation BR, we extract the optimal concepts using a branch and bound algorithm selecting the most suitable elementary relation as defined in section 2.4.. The output of this algorithm is a heap of concepts and a heap of URLs associated with each concept. The heap of concepts is a tree representing a priority queue of concepts where the root of the tree contains the concept with the highest gain. The heap of URLs has the same tree structure where the root contains the URL with the highest rank.

## 5 Development Approach

In this section a tracing of the general search algorithm is introduced via a comprehensive example of searching for the query 'conceptual clustering'. After the process of creating the binary relation of the returned search space is completed, a 2 level hierarchy space of concepts and URLs is displayed. A concept is described as illustrated by a concept no, Gain, name which is the highest ranked keyword associated with the concept and used to describe it, in addition to a list of URLs contained in this concept organized in a heap to provide a high degree of dynamicity.

### 5.1 The User Navigation Design

Our system provides user with 2 ways to navigate through different concepts: the non linear navigation using the tree of concepts and the linear navigation using navigation buttons (next, previous). Each concept is viewed using 2 representations: the hierarchal view of the tree of URLs and a linear view of all URLs in this concept.

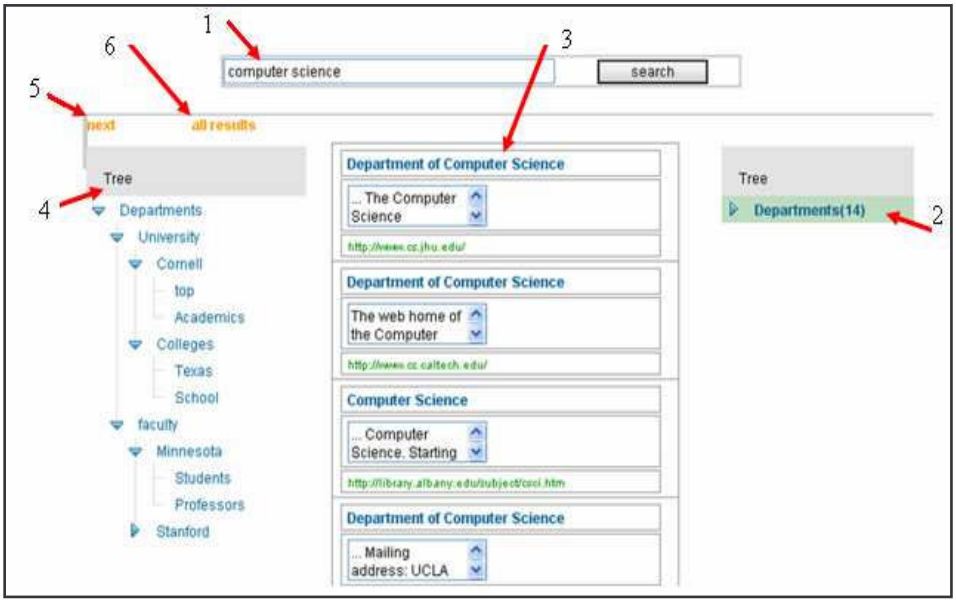



**Fig. 1.** General view of the system interface

### 5.2 Anatomy of the System View

In figure 1 representing the general view of the system interfaces arrows 1, to 6 are described as follows:

1: Search field to enter the query.
2: The tree of concepts with the currently displayed concept highlighted, the tree node represents the concept name and the number of URLs in the concept. If you double-click at (2) a sub-tree of concepts names will appear.
3: Linear view of the URLs that represent the selected concept each with its title, brief description and clickable link to the URL. This could also hold the description of the selected URL from the tree of URLs (4). If you click into a URL, the web page will appear.
4: Tree of URLs associated with the selected concept where the equaled rank URLs are viewed in the same level and the URLs in the leaves have the lowest ranks.
5: Navigation bar that contains 'Next' button which displays the next extracted optimal concept and possibly 'Previous' button (only showed if the current concept displayed is not the first).

6: 'All results' button is used to display the linear representation of all the URLs associated with the current selected concept.

Experience realized by several users showed that the quality of the research pleased to most of them, who appreciate the structure of words representing the main concepts.

## 6 Conclusion

This system employs formal concept analysis as an approach for knowledge discovery and conceptual clustering. A heuristic process of finding coverage of the domain of knowledge was achieved by using the idea of optimal concepts. Our approach is original for meta search engine design. However, ranking function and algorithm efficiency should be improved for a better utilization of the developed system. We should also now explore the incremental version of these algorithms.

**Acknowledgements**: We would like to thank the anonymous reviewers for the useful comments. We are also grateful for Qatar University to support this research work under the grant # 05013CS.

## References

1. Bernhard Ganter and Rudolf Wille: Formal Concept Analysis: Mathematic Foundations. Springer, Berlin/Heidelberg, (1999).
2. Carpineto, C., and Romano, G.,: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. Journal of Universal Computing, (2004) 10, 8, 985-1013.
3. Godin, R., Gecsei, J., and Pichet, C.,: Design of Browsing Interface for Information Retrieval. In N. J. Belkin, & C. J. van Rijsbergen (Eds.), Proc . SIGIR '89, (1989) 32-39.
4. Ganter et al., Bernhard Ganter, Gerd Stumme and Rudolf Wille, editors: Formal Concept Analysis, Foundations and Applications, Volume 3626 of Lecture Notes in Computer Science, Springer, (2005).
5. Wille, R.,: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In I. Rival (Ed.), Ordered sSts. Reidel, Dordrecht-Boston (1982), 445-470.
6. Computer Aided Intelligent Recognition Techniques and Applications, Conceptual Data Classification: Application for Knowledge Extraction. Chapter 23: Ahmed Hasnah, Ali Jaoua, Jihad M.AlJa'am, Editor Muhammad Sarfraz, Wiley, (2005).
7. Jaoua A. and Elloumi S., Galois Connection, Formal Concepts and Galois Lattice in Real Relations: Application in a Real Classifier, The Journal of Systems and Software, 60, (2002), pp. 149-163.
8. Schmidt, and Strohlein: Relation and Graphs, Discrete Mathematics for Computer Scientists. EATCS-Monographs on Theoretical Computer Science. Springer, (1993).