Dynamical modification of context for an iterative and interactive information retrieval process on the web.

Emmanuel Nauer and Yannick Toussaint

LORIA – UMR 7503. B.P. 239, F-54506 Vandœuvre-lès-Nancy cedex, France.
{Emmanuel.Nauer,Yannick.Toussaint}@loria.fr

Abstract. This paper presents an iterative and interactive information retrieval system to search on the web using formal concept analysis (FCA). FCA provides a natural way to organize objects according to their properties and it has been used in recent works to organise in a more convenient manner answers provided by a search engine. The navigation into the lattice helps the user explore a structured and synthetic result. Such a lattice contains concepts that are relevant and some others that are not relevant for a given information retrieval task. We introduce lattices into an interactive and iterative system. The user expresses his negative or positive agreement with some concept of the lattice, in respect with his objective of information retrieval. These user choices are converted into operations over the lattice so to make the context change and to better fit user needs.
Keywords: information retrieval, concept lattices, user feedback.

1 Motivation

Formal concept analysis (FCA) provides a natural way to organise objects according to their properties. In the framework of Information Retrieval (IR), FCA has been applied to query refinement, ranking, documents classification, etc. In a web context, some recent works [5, 1] proposed to built a concept lattice starting from title and snippet words of documents returned by a search engine (like GOOGLE), in order to organise them in a structured and synthetic result. To the best of our knowledge, no work has been done on the evolution of the hierarchy during an IR process. The hierarchy is built once and gives an abstract view of the set of documents, in which the user may navigate. However, there is no possibility to change the lattice which may contain numerous non relevant concepts (and documents). In the same time, relevant documents may exist outside the initial set of documents used to build the lattice. So, making the initial set of documents –and the lattice– evolve is a way to tune the results of the IR system to the user needs.

In the domain of IR, many works have be done to embed the user inside the IR process in order to improve the performances of the systems. In these works, user feedback directly impacts on the IR system. This paper presents the CreChainDo system which plugs the user feedback approach on a lattice navigation. The user explores the lattice, and identifies relevant or irrelevant concepts. The user feedback is converted into a reduction or an extension of the context of the lattice and a new lattice is built.

The paper is organised as follows: section 2 shows how FCA and user feedback improve IR. Section 3 details the CreChainDo system and section 4 gives a concrete example of CreChainDo at work. Future directions for this work conclude the paper.

2 Improving IR using FCA

2.1 Formal concept analysis

	Web	IR	SearchEngine	WebDirectory	Yahoo!	Altavista	Google
d_1	×	×	×	×	×		
d_2	×	×	×				
d_3	×	×	×				
d_4	×			×			
d_5		×	×			×	×
d_6		×	×				

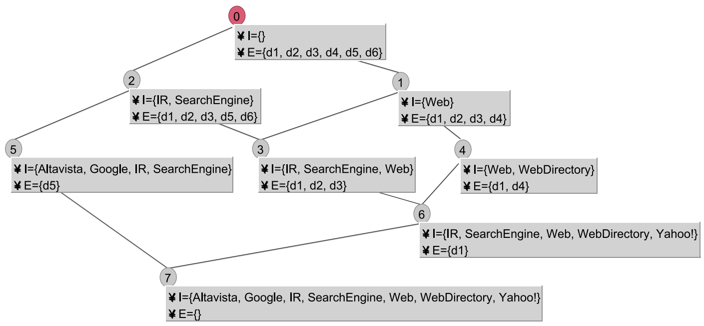

Fig. 1. A binary context and the corresponding concept lattice.

FCA is a mathematical approach to data analysis based on lattice theory. A *formal context* is a triple $\mathcal{K} = (G, M, I)$, where G is a set of individuals (called *objects*), M a set of properties (called *attributes*) and I the relation on $G \times M$ stating that an object is described by a property [11]. Table in the left-hand side of Fig. 1 gives an example of context: G is a set of 6 documents ($\mathtt{d}_1$,...,$\mathtt{d}_6$) and M the set of properties composed of 7 keywords describing the documents. A *formal concept* is a pair (I, E), where E is a maximal set of individuals (called *extent*) and I is a maximal set of properties (called *intent*) shared by this extent. For instance, $(\{\mathtt{IR}, \mathtt{SearchEngine}, \mathtt{Web}\}, \{\mathtt{d}_1, \mathtt{d}_2, \mathtt{d}_3\})$ is a concept (see diagram in the right-hand side of Fig. 1).

Furthermore, the set $\mathcal{C}_{\mathcal{K}}$ of all formal concepts of the context $\mathcal{K} = (G, M, I)$ is partially ordered by extent inclusion also called the *specialization* (denoted $\leq_{\mathcal{K}}$) between concepts. $\mathcal{L} = \langle \mathcal{C}_{\mathcal{K}}, \leq_{\mathcal{K}} \rangle$ is a complete lattice, called the *concept lattice*. The lattice $\mathcal{L}$ can be drawn as a Hasse diagram where nodes are concepts, and line segments specialization links. Fig. 1 illustrates a context and its corresponding lattice. The top concept contains all the documents ; its intent is empty because there is no common property shared by all the documents. On the opposite, the bottom concept is defined by the set of all properties and its extent is empty as none of the documents is described by all the properties. A great number of algorithms has been proposed for the construction of concept lattices, see [11]. For our application, we use CORON, a software platform implementing a rich

Fig. 2. A GOOGLE document returned for the query *"carpineto romano"*.

set of algorithmic methods for symbolic datamining, including concept lattice construction algorithms [19].

2.2 Formal concept analysis for information retrieval

FCA has been applied to solve various core problems of IR systems (a fine state of the art is presented in [5]). Lattice based IR follows the idea that a concept intent can be seen as a query and its extent as the retrieved documents. The underlying idea is that the neighbour concepts of a query concept can be seen as the minimal changes for a query reformulation. The REFINER system [3] exploits these properties, by building only a part of the lattice around the query concept and by displaying it to the user.

Concept lattices are also good candidates for hybrid IR systems based on querying and navigation [6, 8]. The user query is classified into the lattice. This concept is the starting point for a navigation into the hierarchical structure. In this framework, [2] proposes to enhance the lattice structure by exploiting a subsumption hierarchy about the set of terms describing the documents, in the form of a thesaurus. Two specific functionalities for querying and navigating into the lattice are proposed in [9]. The first one, a query-by-example function retrieves documents which own common properties shared by a set of documents. The second one, a similarity measure between concepts (based on their intensions and extensions) ranks similar concepts to a given one.

Finally, lattices are also used for ordering documents returned by an IR system. Approaches have been proposed for document ranking [4], or even for documents re-organisation. The CREDO (standing for Conceptual REorganisation of DOcuments) system [5] (or its adaptation CREDINO for PDA [1]) uses a lattice for clustering and navigating into a set of documents returned by GOOGLE on a given query. Each GOOGLE document, as the one given on Fig. 2, is composed of a title, a snippet and an URL. CREDO builds two binary contexts. The first level of the hierarchy results from the *document* $\times$ *title words* context, whereas the next levels result from the *document* $\times$ *(title + snippet words)* context. In this hierarchy, the intent of a concept is a set of words and its related extent is the set of documents containing all the words of the intent. The initial flat list of documents given by GOOGLE is now organised in a strutured way. The navigation into this hierarchy helps the user to explore the set of documents. We improve the CREDO approach adding interactive capabilities.

2.3 Iterative and interactive information retrieval process

The goal of an IR system is to provide a user with information about a given need. In most IR systems, and especially web search engines, the IR process consists in the submission of a query representing the user need. As a query is a reduce representation of the user's need, some IR systems exploit additional information. Relevance Feedback (RF) [15] provides more information on the search and is known to be effective for improving retrieval accuracy [16].

There are two kinds of RF: *explicit RF* and *implicit RF*. In explicit RF, the user has to give explicitly his feedback to the IR system by, for example, entering keywords, answering to specific questions, marking a subset of retrieved documents as relevant or irrelevant [12], annotating documents [7], etc. In implicit RF, IR systems do not propose any specific interactions for the feedback. The feedback is deduced by the IR system according to all the *implicit* interactions of the user [17]. For example, in a query system, the results returned on a first query may generally not be satisfactory. Often, the user may need to modify his query and to view ranked documents with many iterations before the information need is completely satisfied. All these interactions may be exploited for feedback. Query reformulations may, for example, disambiguate the context of polysemic words [10]. We refer to [13] for a classification of implicit feedback techniques in link with the major works of the domain.

The IR process proposed in CreChainDo implements an explicit RF. The user interacts iteratively with the system for evaluating if a concept is relevant or not. His feedback is linked to a modification of the context used to built the lattice.

3 The CreChainDo system

3.1 Principles

To take into account the iterativity and interactivity of the IR process, CreChainDo [1] integrates a dynamical context modification approach. In the first step of the IR process, the user submits a query and a lattice is built. Then, the user may perform choices over the lattice. Like in IR systems in which the user can accept/reject documents or set of documents, CreChainDo offers the user to select concepts which are relevant and concepts which are not relevant.

Definition 1. *A concept is* relevant *with respect to the user need, if a query Q, resulting from the conjunction of all the words of its intent, makes more precise the user need.*

The underlying idea of this definition is that the lattice do not necessarily contain, at a given time, all the relevant documents that can be found on the web, and that a new search engine interrogation on Q may return new relevant documents. This definition does not take into account the concept extent because the extent of a relevant concept does not necessarily contain only relevant documents.

Definition 2. *A concept is* not relevant *according a user need, if the documents contained in its extent are not relevant for the user need, and if none of the words, defining its intent, can take part of the intent of a relevant concept.*

[1] The experimental protoype is available at http://intoweb.loria.fr/CreChainDo

The CreChainDo interface (see Fig. 4, afterwards) proposes to the user, for each concept of the hierarchy, to click on the icon if relevant and if not relevant. Each user action acts on a context modification and a new lattice is computed. In CreChainDo, the IR process is a succession of context: $K_0 \rightarrow K_1 \rightarrow \ldots \rightarrow K_i \rightarrow K_{i+1} \rightarrow \ldots$. We note $K_i = (G_i, M_i, I_i)$ the context related to the step i. $K_0 = (G_0, M_0, I_0)$ denotes the initial empty context: $G_0 = \emptyset$, and thus $M_0 = \emptyset$ and $I_0 = \emptyset$.

3.2 Context extension

We denote by Q a user query. Q is a non empty set of words : $Q = \{word_i | i > 0\}$. We denote by $DOC(Q)$ the set of documents returned by a search engine to the query Q: $DOC(Q) = \{doc_j | j \geq 0\}$.

Definition 3. *The extension of the context K_i to K_{i+1}, noted $K_i \xrightarrow{+Q} K_{i+1}$, is done by adding the result returned by a search engine to the query Q :*

- *new documents are added to the objects set of the context : $G_{i+i} = G_i \cup DOC(Q)$;*
- *new words are added to the properties set of the context : $M_{i+i} = M_i \cup \{word_i | i \in DOC(Q)\}$;*
- *the relation I_i between G_i and M_i is extended to the relation I_{i+1} between G_{i+1} and M_{i+1}*

In a K_{i+1} *extended* context, objects from G_i are described exactly with the same properties. Therefore, for all $(g, m) \in I_i$, if $g \in G_{i+1}$ then $(g, m) \in I_{i+1}$. For all concept $(g, m_i) \in K_i$, there exists a concept $(g, m_{i+1}) \in K_{i+1}$ where $m_i \subseteq m_{i+1}$. The intent remains unchanged while extent may increase. This ensures a progressive lattice evolution between two steps.

There is two cases, where the context is extended:

- the user submits a new query: the user query Q_0 ($K_0 \xrightarrow{Q_0} K_1$) defines a new context but the user may introduce at anytime in the IR process a new query Q_i (adding new words in the search) between two steps ($K_i \xrightarrow{+Q_i} K_{i+1}$).
- the user evaluates a concept C as relevant for his need: a query composed of the conjunction of all the words defining the intent of C is submitted to a search engine, the context is extended by $K_i \xrightarrow{+intent(C)} K_{i+1}$.

3.3 Context reduction

The objective of a context reduction is to remove from the lattice, concepts that the user evaluates as non relevant for his need.

Definition 4. *The reduction of the context K_i to K_{i+1}, noted $K_i \xrightarrow{-C} K_{i+1}$, is done as follows:*

- *the documents which are in C extent are removed from the context: $G_{i+i} = G_i \setminus extent(C)$;*
- *the words which are in C intent are added to a stopwords list (see 3.4). This implies that $M_{i+i} = M_i \setminus \{word_i | i \in intent(C)\}$;*
- *the relation I_{i+1} between G_{i+1} and M_{i+1} is obtained by deleting into I_i the lines associated to the documents forming $extent(C)$ and the columns with the words of $intent(C)$.*

3.4 Implementation of CreChainDo

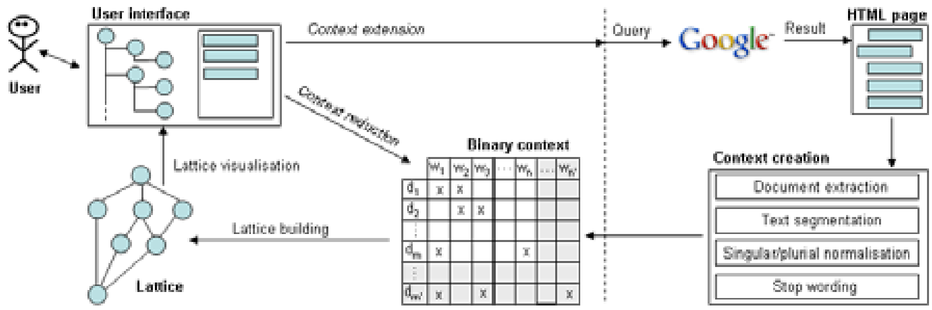

Fig. 3. The CreChainDo architecture.

The architecture of CreChainDo is given on Fig. 3. CreChainDo queries google which returns a web page containing the query result. This page is parsed for extracting a set of documents, as the one given as example on Fig. 2. For each document, the title, the snippet and the URL is extracted. The new documents are added to G_i (the URL is used as key for removing duplicated documents). Creating a context from textual data in CreChainDo follows the classical approach of automatic indexing proposed by [20]. Each document (title ans snippet) is segmented in single words. The singular/plurial inflexional variations are joined using the major rules of variations: a finishing `s` (day → days, etc.), `an` → `en` (man → men, etc.), `y` → `ies` (baby → babies, etc.), `fe` → `ves` (wife → wives, etc.), etc. According to these rules, the vocabulary is normalised by keeping the most frequent form. This avoids the dispersion of the vocabulary. The lattice is directly impacted by reinforcing some concept extensions. Finally, we use the stopword list coming from the Snowball project [18] to eliminate grammatical words (like *"a"*,*"the"*,*"of"*, etc.) and the stopwords list related to negative words identified by the user, to eliminate words that do not represent the user need. All the remaining words are used to create a binary context *document* × *words* (there is no specific word selection). Once the context is built, coron is used to compute the new lattice [19] and a HTML/javascript page is generated and is sent to the web user interface.

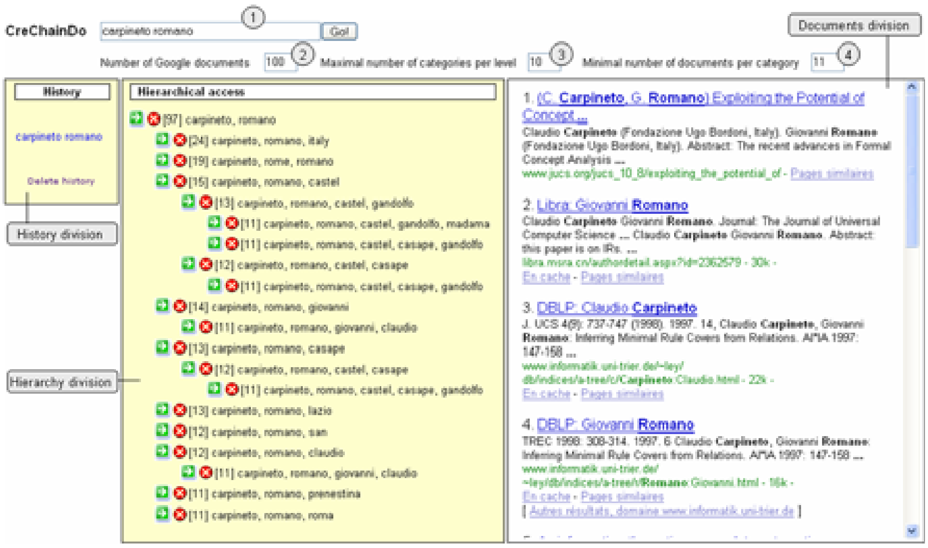

Fig. 4. The CreChainDo user interface, after a request on "carpineto romano".

The user interface (see Fig. 4) is divided in 4 parts:

– on the top, the request interface allows the user to (1) submit a query and to specify some parameters like (2) the number of documents wanted to be returned by GOOGLE. Two parameters related to the hierarchy construction and output are available: (3) the maximal number of concepts (`Mnc`) displayed at each level of the hierarchy and (4) the minimal number of documents (`mnd`) contained in the extent of the concepts (this number can be an expressed by a number, or in percentage of the number of documents of the context).
– on the middle of the interface, the hierarchy resulting from a top-down exploration of the lattice, presents the concept intents in a tree view. We choose that all the concepts of the lattice satisfying the `Mnc` and `mnd` parameters will be displayed.
– on the right-hand side of the interface, a division is designed to display the documents related to the concepts. The content of this division is modified dynamically when the user puts his mouse over a concept of the hierarchy. All the documents being in the extent of the selected concept are displayed.
– on the left-hand side, there is an history division, in which, all the users actions are stored: new query, accepting or rejecting a concept. In the future, the objective of this division is that the user could go back on some of his decisions, as propoposed in the EXALEAD (http://www.exalead.fr/) web search engine.

4 Impact of the user interactions over the hierarchical structure.

This section presents and discuss some experiments. Let say that the user searches for documents about the two italian researchers *Carpineto* and *Romano*. This kind of task is very usual for domain analysis or scientific survey. Let *"carpineto romano"* the first query submitted. The resulting hierarchy obtained by extending the initial empty context K_0 by $K_0 \xrightarrow{+\text{"}carpineto\ romano\text{"}} K_1$ is presented on the left-hand side of Fig. 5. We can notice that the top concept contains only 97 documents instead of 100 requested. The reason is that 3 of the 100 documents returned by GOOGLE do not contain the 2 words in the title nor in the snippet.

4.1 Rejecting a non relevant concept

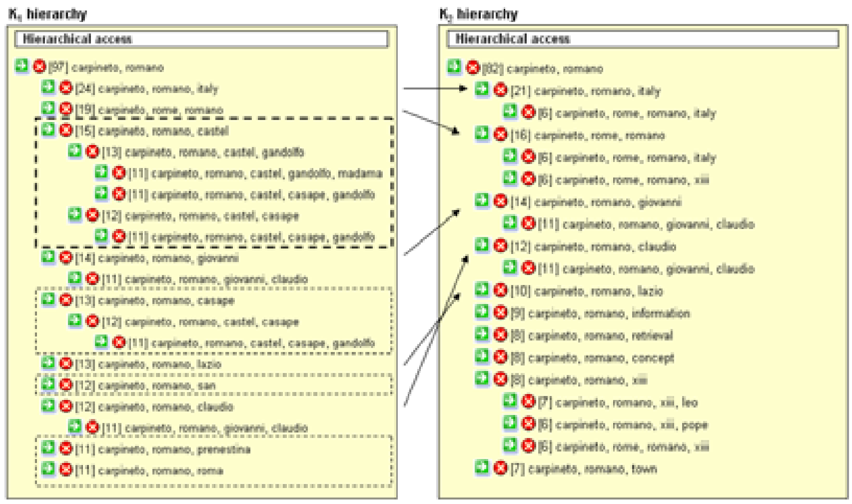

Fig. 5. From K_1 hierarchy to K_2 hierarchy by $K_1 \xrightarrow{-\text{"}carpineto\ romano\ castel\text{"}} K_2$.

The user can quickly see that the hierarchy contains non relevant concepts for a search about *Claudio Carpineto* and *Giovanni Romano*. Let C, with $intent(C) = \{carpineto, romano, castel\}$, be a non relevant concept. Clicking on in front of *"carpineto, romano, castel"* in the user interface produces the K_2 hierarchy presented on the right-hand side of Fig. 5. The K_2 hierarchy contains now only 82 documents. In the tree-view display, the effect is a subtree deletion (but in the lattice, these concepts are either removed or merely adapted if they are linked to another part of the lattice). Eliminating documents and terms of a context can affect the whole hierarchy. As illustrated on the left-hand side of Fig. 5, all the framed subtrees have been deleted and some concept extents have been reduced. For example, 3 documents have been deleted in the extent of the *"carpineto, romano, italy"* concept.

4.2 Accepting a relevant concept

Let C, with $intent(C) = \{carpineto, romano, giovanni, claudio\}$ be a relevant concept. Clicking on in front of C in the user interface produces the K_3 hierarchy presented on Fig. 6. The 87 new documents that have been added produce a more fine grain structuration of the hierarchy.

Fig. 6. Part of the K_3 hierarchy, after $K_2 \xrightarrow{+"carpineto\ romano\ giovanni\ claudio"} K_3$.

4.3 New query submission

Sometimes, during the IR process, the user has to reformulate his need. In CRE-CHAINDO, each new query submitted to the system will continue to add documents onto the context. For example, the part of the hierarchy presented on Fig. 7 concerns an extension of K_3 by a new query on *"carpineto romano lattice"*: ($K_3 \xrightarrow{+"carpineto\ romano\ lattice"} K_4$).

Fig. 7. Part of the K_4 hierarchy, after $K_3 \xrightarrow{+"carpineto\ romano\ lattice"} K_4$.

4.4 Discussion

Three possible types of interaction are available in CreChainDo.

Rejecting a non relevant concept is very useful when a lattice contains concepts which do not immediatly focus on information satisfying the user. This type of lattice is obtained for query with polysemic words, but also if the vocabulary of GOOGLE documents is very scattered. To limit this problem, building concepts on a restricted set of properties –as it is done in CREDO for the first level of the hierarchy– is a possible solution. But in this case, non relevant documents or non relevant concepts are even present into the hierarchy. Therefore, there is a real need to *clean* the hierarchy. Eliminating the noise allows indirectly to focus on relevant documents.

Accepting a relevant concept is a way to add new documents, more specific that those obtained until now, into the context. This is a significant extension of CREDO. Indeed, the hierarchy proposed by CREDO is depth-limited and the degree of specialisation of concepts is limited as well. Indeed, the lattice is built only on 100 documents and the vocabulary used in these documents does not necessarily cover all possible various topics related to the words used to query the system. Moreover, the extent of some concepts can be very small, due to the distribution of the 100 documents over all the concept. In CreChainDo, if a concept C contains few documents –as it is the case for concepts directly subsuming the bottom concept– accepting C will extend the sub-hierarchy starting from C. More specific subconcepts of C will be generated by the construction of a new lattice. So, the depth of the hiearchy and the degree of specialisation are less limited.

Submitting a new query offers the user to reformulate his need by himself, without being restricted on the lattice concepts. So, the user can submit multiple queries to the system and all the results returned by GOOGLE will be synthesised into a same hierarchy. This approach seems to be very interesting for merging results coming from multiple query reformulations and can probably also be applied for merging results coming from multiple information sources, as it is done, for example, by meta search engines.

4.5 Future work

First of all, the iterative approach proposed by CreChainDo, has to be evaluated and validated. More generally, we think that many problems, concerning the IR process, have to be studied concretly. For example: how is it possible to take into account the change of the user need during the process? What are the impacts of our choices for the context modifications on the IR process? For example, the context extension on all the words of a relevant concept is possibly too specific: a context extension restricted on specific words of the relevant concept intent (i.e words which are not inherited from more generic concepts) may be an alternative. These questions, about the CreChainDo IR process, are strongly connected to the lattice construction and the dynamical modification of context.

The *quality* of hierarchies, built according to various strategies like those proposed in CREDO and CreChainDo, has to be examined as well because they play a central role in this type of IR process. We also think that some parts of CreChainDo lattices follows a specific organisation and that the study of the lattice properties is a way to improve the hierarchy quality. For example, in Fig. 6, the concept *"[99] carpineto, romano, giovanni, claudio"* is subsumed by the two concepts *"[101] carpineto, romano, giovanni"* and *"[103] carpineto, romano, claudio"*. We agree that a multiple inheritance favors the access to a concept, which is thus reachable by several paths. Nevertheless, the interest of the two concepts *"[101] carpineto, romano, giovanni"* and *"[103] carpineto, romano, claudio"* can be discussed with regards to (1) their connexions with the *"[99] carpineto, romano, giovanni, claudio"* one and of (2) the very high similarity of their extents (containing *[99]*, *[101]* and *[103]* documents). This difference mainly results from the exploitation of the snippets, which boundaries are not wide enough to cover all the 4 words (but these missing words appear in the complete document).

Finally, an attractive direction concerns the potential of synthesis of lattice for a set of search engine results. This approach can be applied for a same query submited to multiple search engines, or also for multiple queries. In the framework of CreChainDo, all the user interactions which are stored in the history could be exploited for an implicit RF [14]. According to the set of positive words (extracted from user queries and relevant concepts) and to the set of negative words (extracted from user non relevant concept), multiple queries can be submitted to GOOGLE. These queries could automatically be generated from all the possible conjunctions of positive words and from a filtering on all the negative words.

5 Conclusion

The CreChainDo system, presented in this paper, is an innovative application of FCA for IR on the web. The user feedback approach has been plugged on the exploration of a lattice used to structure a flat answers list of documents returned by a search engine. With CreChainDo, the user can act directly on the lattice, to make it evolve by a dynamical modification of context. Cleaning the lattice, extending it in a specialised direction or using the lattice as a merging tool for multiple queries offer significant possibilities for IR. This approach can also be seen as a web mining one, in which the result of a step is exploited on the next step.

CreChainDo extends FCA for IR in a dynamical way. Lattices may evolve during the IR process. The user is not restricted anymore by a static and once computed structure.

References

1. C. Carpineto, A. Della Pietra, S. Mizzaro, and G. Romano. Mobile clustering engine. In *Proceedings of the 28th European Conference on Information Retrieval,*

ECIR 2006, volume 3936 of *LNCS*, pages 155–166. Springer, 2006.
2. C. Carpineto and G. Romano. Information retrieval through hybrid navigation of lattice representations. *International Journal of Computer Human Studies*, 45(5):553–578, 1996.
3. C. Carpineto and G. Romano. Effective Reformulation of Boolean Queries with Concept Lattices. In *Flexible Query Answering Systems, Third International Conference (FQAS'98)*, volume 1495 of *LNCS*, pages 83–94. Springer, 1998.
4. C. Carpineto and G. Romano. Order-Theoretical Ranking. *Journal of the American Society for Information Science*, 51(7):587–601, 2000.
5. C. Carpineto and G. Romano. Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. *Journal of Universal Computer Science*, 10(8):985–1013, 2004.
6. R.J. Cole, P.W. Eklund, and G. Stumme. Document retrieval for email search and discovery using formal concept analysis. *Journal of Applied Artificial Intelligence (AAI)*, 17(3):257–280, 2003.
7. P. A. Dmitriev, N. Eiron, M. Fontoura, and E. Shekita. Using annotations in enterprise search. In *Proceedings of the 15th International Conference on World Wide Web*, pages 811–817, 2006.
8. J. Ducrou. DVDSleuth: A Case Study in Applied Formal Concept Analysis for Navigating Web Catalogs. In *Conceptual Structures: Knowledge Architectures for Smart Applications, 15th International Conference on Conceptual Structures (ICCS 2007)*, volume 4604 of *LNCS*, pages 496–500. Springer, 2007.
9. J. Ducrou, B. Vormbrock, and P. W. Eklund. FCA-Based Browsing and Searching of a Collection of Images. In *Conceptual Structures: Inspiration and Application, 14th International Conference on Conceptual Structures (ICCS 2006)*, volume 4068 of *LNCS*, pages 203–214. Springer, 2006.
10. S. T. Dumais, E. Cutrell, R. Sarin, and E. Horvitz. Implicit queries (iq) for contextualized search (demo description). In *Proceedings of SIGIR 2004*, page 594, 2004.
11. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer, Berlin, 1999.
12. R. Kassab, J.-C. Lamirel, and E. Nauer. Novelty detection for modeling user's profile. In *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference*, pages 830–831, California, 2005. AAAI Press.
13. D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.
14. S. Y. Rieh and H. Xie. Analysis of multiple query reformulations on the web: the interactive information retrieval context. *Information Processing and Management*, 42(3):751–768, 2006.
15. J.J. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323, 1971.
16. G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *JASIS*, 41(4):288–297, 1990.
17. X. Shen, B. Tan, and C. Zhai. Context-sensitive information retrieval using implicit feedback. In *Proceedings of SIGIR'05*, 2005.
18. Snowball project web site. http://snowball.tartarus.org/, visited june 2007.
19. L. Szathmary and A. Napoli. CORON: A Framework for Levelwise Itemset Mining Algorithms. In *Supplementary Proceedings of The Third International Conference on Formal Concept Analysis (ICFCA '05), Lens, France*, pages 110–113, 2005.
20. C. J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.