Using Formal Concept Analysis for Heterogeneous Information Retrieval

Ibtissem Nafkha[1] and Ali Jaoua[2]

[1]University of Tunis, Department of Computer Science,
Campus Universitaire,le Belvédère, 1060, Tunis, Tunisia.
[2]University of Qatar, Faculty of Sciences,
Department of Computer Science, Doha, Qatar.
ibtissem.nafkha@fst.rnu.tn, jaoua@qu.edu.qa

Abstract. With the advent of the Web along with the unprecedented amount of information coming from sources of heterogeneous data, Formal Concept Analysis (FCA) is more useful and practical than ever, because this technology addresses important limitations of the systems that currently support users in their quest for information. In this paper, we will focus on the unique features of FCA for searching in distributed heterogeneous information. The development of FCA-based applications for distributed heterogeneous information returns a major gain.

1 Introduction

The information systems these days manage, import, broadcast, exchange and integrate big volumes of sometimes recorded data, often in different formats (documents, cards, tables). With the internet development, the institutions are often confronted to the manipulation and the analysis of important information volumes. These informations are often coming from heterogeneous data sources and are themselves of heterogeneous nature. Regarding this heterogeneity, the integration or the simple exchange of the data is not an easy task if the different intervening (producers or information consumers) do not agree on the semantic of data. It is therefore very difficult to research the answer to an information need in all bases.

In this direction, we are very interested in defining an approach that is focused particularly on the detection of the similar objects. Furthermore, the important volume that occupies the heterogeneous data creates gaps and technical difficulties such as pertinent information deficiency and the loss time for precise information research. In this context, we propose an analysis and an interpretation approach of the similar objects allowing jointly to realize a more effective research and to extract automatically the information from the dispersed sets of heterogeneous data in the framework of the cooperative work. Our approach is based on the formal concept analysis.

Radim Bělohlávek, Václav Snášel (Eds.): CLA 2005, pp. 107–122, ISBN 80–248–0863–3.

So, this paper is organized as follows. In section 2, we introduce some basic definitions on formal analysis. Then in section 3, we present the related work. Section 4 is devoted to the presentation of proposed system for searching in heterogeneous information. In section 5 and 6, we present the evaluation of our system.

2 Mathematical Foundations

Among the mathematical theories recently found with important applications in computer science, lattice theory has a specific place for data organization, information engineering, data mining and for reasoning. It may be considered as the mathematical tool that unifies data and knowledge or information retrieval [1,4,7,10,18,20,23]. In this section, we define formal context, formal concept, Galois connection and the lattice of concepts associated to the formal context.

2.1 Formal Context

Definition 1. A formal context is a triple k = <O,P,R>, where O is a finite set of elements called objects, P a finite set of elements called properties and R is a binary relation defined between O and P. The notations (g,m), or R(g,m)=1, mean that "formal object g verifies property m in relation R" [3,12].

Example 1. Let O = {a_1, a_2, a_3, a_4, a_5, a_6} be a set of person of different grade and P = {b_1, b_2, b_3, b_4, b_5, b_6, b_7} be a set of the properties. This context describes the professional qualifications verified by the persons set according to the binary relation R. The

	b_1	b_2	b_3	b_4	b_5	b_6	b_7
a_1	1	0	1	0	0	0	0
a_2	1	1	0	0	0	0	0
a_3	1	1	1	1	0	0	0
a_4	1	1	1	0	1	0	0
a_5	1	0	1	0	0	0	1
a_6	1	1	0	0	0	1	1

Table 1. An example of a formal context.

2.2 Galois Connection

Definition 2. Let A $\subseteq$ O and B $\subseteq$ P two finite sets, R a relation on O x P. For both sets A and B, operators *f*(A) and *h* (B) are defined as [12]:

$$f(A) = \{m \mid \forall g, g \in A \rightarrow (g,m) \in R\}$$

$h(B) = \{g \mid \forall m, m \in B \rightarrow (g,m) \in R\}$

Operator *f* defines the properties shared by all elements of A. Operator *h* defines objects sharing the same properties included in set B. Operators *f* and *h* define a Galois Connection between sets O and P [12].

Proposition 1. Operators *f* and *h* define a Galois connection between O and P, such that if A1, A2 are subsets of O, and B1, B2 are two subsets of P, then f and h verify the following properties [12]:

- $A_1 \subseteq A_2 \Rightarrow f(A_1) \supseteq f(A_2)$
- $B_1 \subseteq B_2 \Rightarrow h(B_1) \supseteq h(B_2)$
- $A_1 \subseteq h\ o\ f(A_1)$ and $B_1 \subseteq f\ o\ h(B_1)$
- $A \subseteq h(B) \Leftrightarrow B \subseteq f(A)$
- $f = f\ o\ h\ o\ f$ and $h = h\ o\ f\ o\ h$

2.3 Formal Concept

Definition 3. A formal concept of the context <O,P,R> is a pair (A,B), where $A \subseteq O$, $B \subseteq P$, such $f(A) = B$ and $h(B) = A$. Sets A and B are called respectively the domain (extent) and range (intent) of the formal concept [3,12].

2.4 Concept Lattice

Definition 4. From a formal context <O,P,R>, we can extract all possible concepts. In [12], we prove that the set of all concepts may be organized as a lattice, when we define the following partial order relation << between two concepts, (A1,B1) << (A2,B2) $\Leftrightarrow$ (A1 $\subseteq$ A2) and (B2 $\subseteq$ B1). The concepts (A1,B1) and (A2,B2) are called nodes in the lattice.

2.5 Objects Similarity

The object similarity can be envisioned according to two view points:
- The semantic view point: the objects are similar if they have commons properties,
- The system view point: to take into account the object model have vector model.

Semantic Similarity. Definition 5. Let k=<O,P,R> a formal context, O is object set, P is properties set and R is the binary relation between O and P. The similarity between two objects *a* and *b* is considered the commons properties. Let *a* and *b* two elements of O, P_a the verifying properties by the object *a* and P_b the verifying properties by the object *b*. The commons properties between two objects *a* and *b* forms the set $P_a \cap P_b$. The similarity between two objects is calculated with the following formula [23]:

$$\text{Similarity (a, b)} = \frac{\left|P_a \cap P_b\right|}{\left|P_a \cup P_b\right|} \quad (1)$$

The similarity is a value in the interval [0,1]. In our system, we use this formula in order to detect the similar documents.

Example 2. Let two formal contexts, presented in table 2, defined respectively between 5 objects {O_1, O_2, O_3, O_4, O_5} and three properties {A, C, D} and between four objects {O_6, O_7, O_8, O_9} and four properties {A, B, C, E}.

	A	C	D
O_1	1	1	1
O_2	1	1	0
O_3	1	0	1
O_4	1	0	0
O_5	1	1	1

	A	B	C	E
O_6	0	1	1	1
O_7	1	1	1	0
O_8	1	1	0	1
O_9	0	1	0	0

Table 2. Formal context example.

The object O_1 is similar to the object O_6 with similarity degree equal to 0.2. Indeed, objects O_1 and O_6 verify in five different properties which one is common. The similarity between O_1 and O_6 is:
Similarity (O_1, O_6) = 1 / 5 = 0.2

System similarity. In order to measure the similarity between two objects *a* and *b*, it necessary to take in consideration the different object models. For this reason, we present only the similarity calculation between two objects in the vector seen model the complexity of the others model. [11,21,22,24,25,26,27]

Definition 6. The similarity between two objects a and b in the vectorial model [24, 25, 26] is measured as the angle cosines between two vectors presenting those objects.

$$\text{Similarity (a, b)} = \cos(\vec{a}, \vec{b}) = \frac{\vec{a}.\vec{b}}{\left\|\vec{a}\right\|.\left\|\vec{b}\right\|} \quad (2)$$

Object Similarity Choice. We mention that the object similarity value whatever the view point system or semantic is a value in the interval [0,1].

This object similarity criterean may crold two values: two objects may seen alike or different from each other. So, we determine two sets *Sim_objet* and *Dis_objet* according to similarities and dissimilarities of an object with an object *a*:

$$Sim\_objet\ (a) = \{\ b\ ;\ Similarity\ (a,b) >= \alpha_{sim}\ \}$$
$$Dis\_objet\ (a) = \{\ b\ ;\ Similarity\ (a,b) < \alpha_{sim}\ \}$$

where α_{sim} is the threshold that determines the object notion near or distant. In our work, this threshold is provided by the user. The given value of the research session means that the user accepts the similar answers with this degree. Seen that we use the concepts formal analysis as basic foundation of our research approach, we do not consider the similarity from the system view point but we are very interested in the similarity from the semantic view point.

3 Related Work

Using FCA can complement the existing search systems to address some of their main limitations. Basically, FCA exploits the similarity between documents in order to offer an automatic support structure (i.e., the document lattice) in which we place the information retrieval process. The document lattice can be used to improve basic individual search strategies [1,2,4,13]. Moreover, query refinement is one of the most natural applications of concept lattices. Its main objective is to recover from the null-output or the information overload problem. The concept lattice may be used to make a transformation between the representation of a query and the representation of each document [5,6,7,8,9]. The query is merged into the document lattice and each document is ranked according to the length of the shortest path linking the query to the document concept. On the other hand, in the set of terms describing the document, there exist hierarchies in the form of thesaurus [4,10,13,14]. The information search using FCA takes as input a query that will be forwarded to a selected search engine [6,7,8]. The first pages retrieved by the search engine in answer to the query are collected and parsed. At this point, a set of index units that describe each returned document is generated; such indices are next used to build the concept lattice corresponding to the retrieved results. The last step consists in showing the lattice to the user and managing the subsequent interaction between the user and the system. In spite of such limitations such as for larger information collection, generally we get a huge number of reference, we are interested in building a FCA-based system for distributed information, which may affect both the efficiency and the effectiveness of the overall system [18,19,20]. These systems suppose that a same document is identified in same manner that presents a strong hypothesis. In order to reduce this constraint, we proposed a similar object detection method. While basing itself on this last one, we have defined a cooperative system of heterogeneous information retrieval *HIC2RS* that will be described in the next section.

4 Cooperative Conceptual Retrieval System for Heterogeneous Information

We present in this section the cooperative research for heterogeneous information. While considering the formal concept analysis as mathematical foundation, we propose an heterogeneous information conceptual cooperative retrieval system *HIC2RS*, as illustrated in figure 1, that is composed of two parts:

1) The first part is the cooperative information retrieval system handling local databases. The search of the answer to a query consists in applying a research conceptual approach on every local database. As a result, we will have concepts set forming the content of a *Response* vector.

2) The second part is the final answer formulation that operates in two steps :

   i) *Similar objects detection* based on the *Response* vector and on the local databases set, and

   ii) The concepts merger based on the similar objects set and operated according to the similarity threshold given by the user in order to offer the final answer.

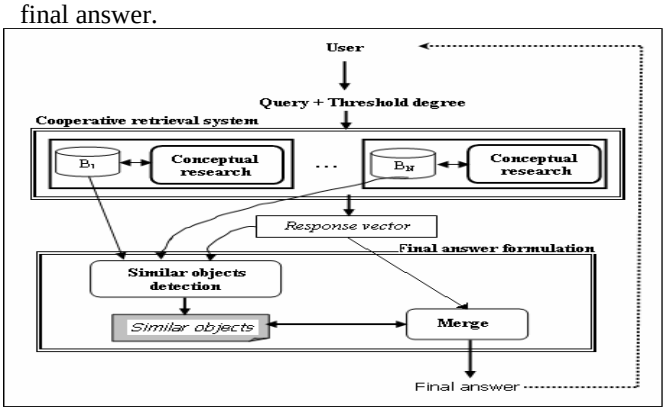

Fig. 1. Heterogeneous Information Cooperative Conceptual Retrieval System Architecture.

4.1 Cooperative Information Retrieval System

The first part of the system *HIC2RS* is formed of information retrieval systems set that cooperate to give the complete answer to a query. Every information retrieval system has access to a local database on which it applies the Galois connection to rediscover the satisfactory documents query. This last one is keywords set. To resolve a query (Qr), every conceptual information retrieval system executes the research algorithm, presented in the following, on its local database (LD). This application gives us concepts set forming the *Response* vector (RV).

```
Algorithm Research
Inputs: Query: Qr
         Local database: LD
Output: Response Vector: RV
Begin
     M := the keywords of LD.
     Ml := M ∩ Qr
     RV contains the concept obtained by Galois connec-
     tion application on Ml.
End
```

4.2 Final Answer Formulation

In this section, we present the second part of the system *HIC2RS* that is the final answer formulation. The final answer formulation is carried out in two steps: i) the detection of the similar objects of the *Response* vector, and ii) the merger of the different answers based on the *Response* vector and on the similar objects. The final answer formulation consists in the application of the algorithm *Merge_IH* that we propose on the *Response* vector basing on the query and on the similarity threshold to have the final answer.

Similarity Objects Detection. The similar objects detection consists in examine the documents that figure in the *Response* vector and calculating the similarity between them. From the concepts, we create a similar objects set. This set contains the similarity degrees between the different documents. The similarity degree calculation between two documents is based on the formula (1) defined in section 2.5. In fact, seen that our system is based on the terminologies of the concepts formal analysis, it is useless to use the similarity from the system view point that depends on used model to present and search the information such as the vectoriel model. While taking account of the keywords number of every document and the number of common keywords between them, the similarity degree between two documents is calculated.

Answer Merge. Basing on the calculated similarity degrees as well as on the *Response* vector concepts, we formulate the final answer to the query. The merger is based on algorithm *Merge_IH* that we propose by the continuation.

This merger algorithm combines the *Response* vector concepts while respecting certain conditions. We construct the final answer in a repeated way. Initially, the final answer is an empty set. We treat the concepts set element by element.

For every element, if the keywords (the extension) of the concept are different of those of query, we add then the documents (his intention) to the final answer. If this condition is not satisfied, we search the similar documents to those of other concepts of the *Response* vector (the intention) verifying the threshold similarity, and we calculate the union of the extensions (to obtain the under together keywords). We continue to construct these sets of similar documents until we find all the query keywords.

This algorithm has as entry the query, the similarity threshold and the *Response* vector and as a result the final answer.

```
Algorithm Merge_IH
    Inputs : Query: Qr
         Response Vector RV: a concepts set C1 .. CN
        Threshold similarity: S
    Output : Final answer: FA
Begin
   FA := ∅•// initialize the final answer
    For each concept Ci of RV do
      If extent of concept Ci = Qr then
         Add the intent of Ci to FA
       Else
         While exist a concept Cj (j > i) do
            - Initialize P by the extent of Ci
          - Initialize D by the intent of Ci
            While (P <> Qr and exist a concept Cj) do
                - Add the extent of Cj to P
                - Search the similar documents, with
                the threshold S, between the intent of
                the concept Cj and the elements of D:
                      D := Similar (D, intent_ Cj,S)
                - Pass to the next concept
             End do
            If (P = Qr ) then
              Add D to FA : FA :=FA ∪ D
             End if
           End do
       End if
   End for
End.
```

The similar function consists in looking the similar objects with a similar threshold in two objects sets. This research is based on the similar objects set found at the time in the phase of the similar objects detection. We keep only the objects having a similarity degree greater than the similarity Threshold. The function is described in the following and it has as inputs two objects sets A1 and A2 and a similarity threshold α and as output the set A3.

```
Function Similar
Inputs:
  Objects sets: A1, A2.
  Similarity Threshold: S
Output: Objects set: A3
Begin
    A3 := Ø
    For each object di of A1 do
```

```
      For each object dj of A2 do
       - Calculate the similarity between two objects di
       and dj:
                α := |P di ∩ P dj| / |P di ∪ P dj|
       - If α>=S, add objects di and dj and the similarity
       α to A3.
          End if
       End for
      End for
      Return (A3)
End.
```

4.3 Illustrative Example

We take an illustrative example to show the *HIC2RS* system functionalities. Let the databases presented in tables 2, 3 and 4. These databases describe documents set indexed by a keywords set. For the query: "Which documents indexed by the keywords M_2, M_3 and M_4 having a similarity Threshold 0.33", the query is formed by three keywords M_2, M_3 and M_4. The treatment of this query is carried out in two steps.

- **Step 1 : Cooperative Research**

The research principle is explained in figure 2.

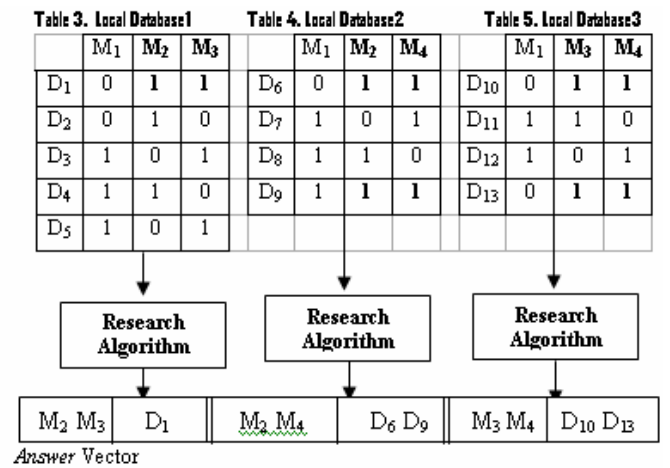

Fig. 1. Cooperative Information Retrieval System.

Every conceptual information retrieval system applies *algorithm retrieve* on its local database. The Galois connection application on the query keywords sets existing in the first database presented in table 3 and the query ($M_1 = \{M_2,M_3\}$) gives the documents set $\{D_1\}$. The found concept is then ($\{M_2,M_3\}$, $\{D_1\}$).

For the second local database presented in table 4, the Galois connection application for the keywords M_2 and M_4, the common found keywords between the local database keywords and those of the query, we give the documents $\{D_6, D_9\}$. So, the result for this local database is formed by the concept ($\{M_2,M_4\}$, $\{D_6,D_9\}$).

The third local database presented in table 5 contains the keywords M_3 and M_4. While applying the Galois connection, we find the documents set $\{D_{10},D_{13}\}$. Thus, the concept ($\{M_3,M_4\}$, $\{D_{10},D_{13}\}$) is the result of this research.

We obtain three concepts from different local databases that we find in the *Response* vector presented by the table 6.

1		**2**		**3**	
M_2 M_3	D_1	M_2 M_4	D_6 D_9	M_3 M_4	D_{10} D_{13}

Table 3. The*Response* vector.

Basing ourselves on this vector, we construct the final answer.

- Step 2: Final Answer Formulation

The final answer formulation is realized in two phases: similar objects detection and the answers merger.

<u>*Similar objects detection*</u> : The *Response* vector contains three concepts that we examine one by one. The first concept contains the document D_1. We calculate then the degree of similarity between this document and every document existing in the two other concepts that are D_6, D_9, D_{10} and D_{13}. The same treatment is carried out on the document D_6. We calculate the similarity degree between D_6 and D_{10} then between D_6 and D_{13}. The same treatment is done on the document D9. The degrees of calculated similarities are the following ones:

Similarity(D_1, D_6*) = 1/3 =0.33; Similarity (*D_1, D_9*)= 1/4 = 0.25;*

Similarity (D_1,D_{10}*) = 1/3 = 0.33; Similarity (*D_1,D_{13}*) = 1/3 = 0.33;*

Similarity (D_6,D_{10}*) = 1/3 = 0.33; Similarity (*D_6,D_{13}*) = 1/3= 0.33;*

Similarity (D_9,D_{10}*) = 1 / 4 =0.25; Similarity (*D_9,D_{13}*) = 1/4 = 0.25;*

<u>*Answer Merge*</u> : We remind that our query is $\{M_2,M_3,M_4\}$ and the similarity threshold is 0.33. Initially, the final answer is an empty set. We treat the first concept of the Response vector. Its keywords are different from the query. So, we merge those keywords with those of the second concept and we search the similar documents. The result of this research is the documents set $\{D_1,D_6\}$, considering that the documents D_1 and D_6 are similar with the degree 0.33, and that the keywords union is the set $\{M_2,M_3,M_4\}$ that is equal to the query. The similarity between D_1 and D_9 is equal to

0.25 that is less than the similarity threshold. So, we ignore D_9 and we add the found documents to the final answer. At this step, the final answer is the set $\{D_1,D_6\}$.

Then, we calculate the union of the keywords and the similar documents between the first and the third concepts of the *Response* vector. The merge result is the set $\{M_2,M_3,M_4\}$ that is equal to the query. We remark that the documents D_{10} and D_{13} are similar to D_1 and to D_6 with the degree superior to 0.33. So, we add those documents to final answer that becomes $\{D_1,D_6,D_{10},D_{13}\}$.

Thus, we continue with the next concept. We merge the keywords of the second and the last concepts. The result is the set $\{M_2,M_3,M_4\}$. The similar documents are $\{D_6,D_{10},D_{13}\}$ that we add to the final answer. The final answer is now the set $\{D_1,D_6,D_{10},D_{13}\}$ that will be delivered to the user.

Remark 1: If we take for example a similarity threshold equal to 0.8, our system returns an empty answer. This answer explains oneself by the fact that there doesn't exist similar objects for this degree. As opposed to the threshold equal to 0.2, the final answer is then composed by all documents forming the *Response* vector. This can be explained by the fact that the similarity degrees between the different documents are greater than the given value. Thus, our approach considers that the documents set represent the same knowledge and we evade late the empty answers.

5 Complexity Analysis

In order to evaluate the system *HIC2RS*, we calculate the temporal and the spatial complexities.

5.1 Temporal Complexity

We suppose that a database has *n* objects and *m* properties and we dispose of *k* local databases.

We recall the steps of our system *HIC2RS*:

- Phase 1: the concepts research from the different local databases.

- Phase 2: the similar objects detection and the merge of *k* found concepts.

The temporal complexity C_T of the system is then:

$C_T = C_{Phase\ 1}(n,m,k) + C_{Phase\ 2}(n,m,k)$

The phase 1 needs $k \times n \times m$ operations and the phase 2 needs $k \times (k-1)/2 + (n \times k)$ operations. So, the temporal complexity is: $C_T = k \times n \times m + k \times (n+1) + (n \times k) = (k \times n \times m) + (k^2 -$

$k)/2+n\times k \approx O(k\times n\times m+k^2)$ operations. The temporal complexity of the system *HIC2RS* is then in order of $O(k\times n\times m+k^2)$ operations.

5.1 Spatial Complexity

The system *HIC2RS* uses *k* matrix of *n* lines and of *m* columns, a vector of *k* elements as well as a square of dimension *n*. The system reserves thus $(k\times n\times m)+k+(n\times n)$ memory cases. So, the spatial complexity of the system *HIC2RS* is equal to: $C_S = (n\times m\times k+ k+n^2)$.

6 Evaluation

The system *HIC2RS* treats heterogeneous information. Indeed, to remedy the problem of the existence of different identifications for similar or identical documents, we proposed a similar objects detection method during the cooperative information retrieval process. The implementation of this system consists first of in fragmenting a test collection and next in releasing the retrieval process while supposing that a same document can have different identifications. This hypothesis is based on unit *similar objects detection*. The experiment was conduced on CRAN and MED collections. The CRAN collection (Cranfield collection) includes a textual corpus that has a size upper than 1.6Mo. This collection contains 1400 documents and 4612 different terms and it is tested on 225 queries. The MED collection includes a textual corpus that has a size upper than 1.1Mo. It contains 1033 scientific articles extracted from the medicine database domain and 5831 different terms and it is tested on 30 queries. With experiments done on the MED and CRAN test collections, we noticed that the final quality of retrieval improved in term precision and recall that in term answer times.

The figure 3 illustrates the precision and recall graph of the MED test collection for the system treating homogenous information CIRS and HIC2RS.

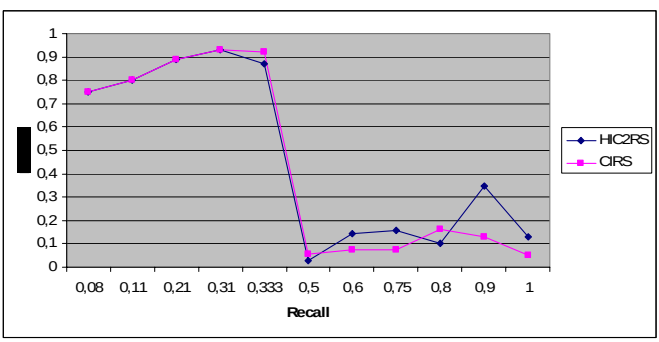

Fig. 2. Precision and recall graph for the MED test collection.

We note, according to figure 3, that for the MED test collection, the measure of average precision has 11 reminder points for the system treating information homogenous (*CIRS*) is in the order of 43.9%. While, for the system *HIC2RS* treating information heterogeneous is on the order of 46.7%. Thus, the similar object detection integration gives an improvement of average precision on the order of 6.4%.

All the same, experimentations done on the CRAN test collection fragmented showed an improvement of average precision of the CRAN test collection on the order of 7.5%. (figure 4).

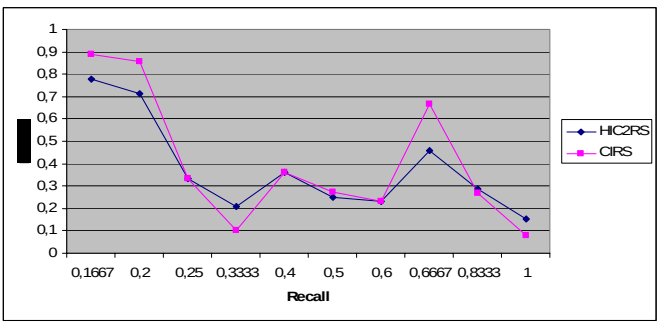

Fig. 3. Precision and recall graph for the CRAN test collection.

The figure 5 shows that *HIC2RS* treats different MED test collection queries faster than the conceptual information retrieval system.

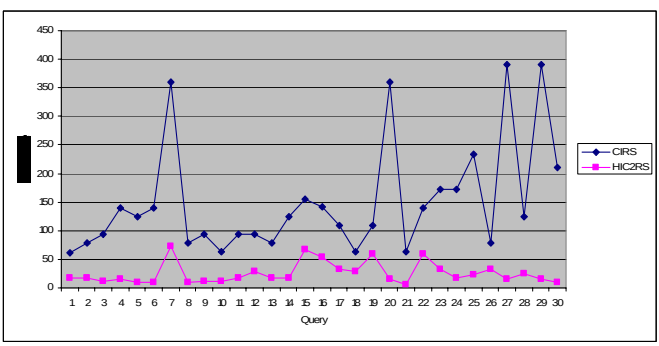

Fig. 4. The answer time take by the systems *CIRS* and *HIC2RS* for the MED test collection.

Of even for the CRAN test collection, the answer time take by *HIC2RS* is lower than the one take by the system treating information homogenous (to see figures 6).

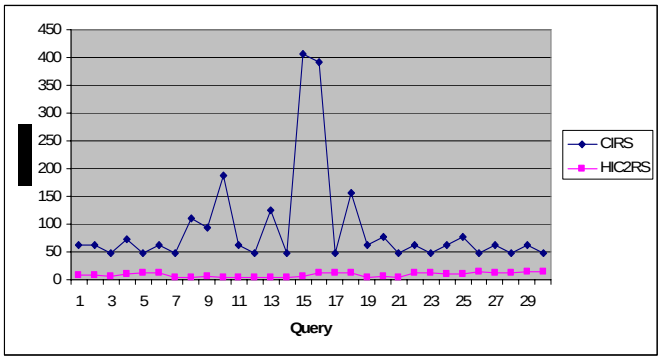

Fig. 5. The answer time take by the systems *CIRS* and *HIC2RS* for the *CRAN* test collection.

7 Conclusion

We presented in this paper a conceptual cooperative retrieval system for heterogeneous information (*HIC2RS*). Being given a heterogeneous environment constituted by a set of information retrieval systems handling each a local database, our approach allows soliciting these databases in order to have a complete answer to a user query. In fact, after a query and according to a similarity threshold given by the user, our system releases conceptual research processes on the different local databases and it will have as a result a concepts set. Basing on this concepts set and on the similarity threshold, the system formulates the final answer that it delivers to the user. The similar objects detection method, that we defined, enriched the returned answers of different databases. This method improved average precision of 6.4% for the MED test collection and of 7.5% for the CRAN test collection.

References

1. Aboud M., Chrisment C., Razouk R., Florence S., Soulé-Dupuy, Query a Hypertext Information Retrieval System by use of Classification. *Information Processing and Management*, 29(3), (1993) 387-396.

2. Amati G., Carpineto C., and Romano G., FUB at TREC-10 Web Track: A Proabilistic Framework for Topic Relevance Term Weighting. *In Proceedings of the 10th Text REtrieval Conference (TREC-10)*, NIST Special Publication 500-250, Gaithersburg, MD, USA (2001) 182-191.

3. Bordat J.P., Calcul pratique du treillis de Galois d'une correspondance. *Math. Sci. Hum.*, 96, (1986) 31-47.

4. Carpineto C. and Romano G., Using Concept Lattices for Text Retrieval and Mining. *In the 1st International Conference on Formal Concept Analysis*, Darmstadt, Germany, (2003).

5. Carpineto C. and Romano G., Information retrieval through hybrid navigation of lattice representations. *International Journal of Human-Computer Studies*, 45(5), (1996) 553-578.

6. Carpineto C. and Romano G., A lattice conceptual clustering system and its application to browsing retrieval. *Machine Learning*, 24(2), (1996) 1-28.

7. Carpineto C. and Romano G., Effective reformulation of Boolean queries with concept lattices. *In Proceedings of the 3rd International Conference on Flexible Query-Answering Systems*, pages 83-94, Roskilde, Denmark, 1998.

8. Cole R. and Eklund P., Browsing semi-structured web texts using formal concept analysis. *In Proceedings of the 9th International Conference on Conceptual Structures*, Stanford, CA, USA, (2001) 319-332.

9. Efthimiadis E., Query expansion. *In M. E. Williams, editor, Annual Review of Information Systems and Technology*, v31, American Society for Information Science, Silver Spring, Maryland, USA, (1996) 121-187.

10. Ferrfie S. and Ridoux O., A file system based on concept analysis. *In Proceedings of the 1st International Conference on Computational Logic*, London, UK, (2000) 1033-1047.

11. Fuhr and C. Buckley, A probabilistic learning approach for document indexing, *ACM Transactions on Information System* 9, 19991, N°3, pages 223-248.

12. Ganter B. and Wille R., Formal Concept Analysis - Mathematical Foundations. *Springer*, 1999.

13. Godin R. and Mili. H., Building and Maintaining Analysis Level Class Hierarchies Using Galois Lattices. *In Proceedings of the 8th Annual Conference on Object Oriented Programming Systems Languages and Applications*, Washington, D.C., USA, (1993) 394-410.

14. Godin R., Missaoui R., and April A., Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38: (1993) 747-767.

15. Godin R. , Saunders E. , and Jecsei J., Lattice model of browsable data spaces. *Journal of Information Sciences*, 40: (1986) 89-116.

16. Jaoua A., Bsaies Kh., and Consmtini W., May reasoning be reduced to an Information Retrieval problem. *Relational Methods in Computer Science*, Quebec, Canada, (1999).

17. Jaoua A., Al-Rashdi A., AL-Muraikhi H., Al-Subaiey M., Al-Ghanim N., and Al-Misaifri S., Conceptual Data Reduction, Application for Reasoning and Learning. *The 4th Workshop on Information and Computer Science*, KFUPM, Dhahran, Saudi Arabia, (2002).

18. Nafkha I., Elloumi S. and Jaoua A., Conceptual Cooperative Information Retrieval System. *In International Arab Conference on Information Technology*, Doha December 16-19, Qatar, (2002) 534-539.

19. Nafkha I., Elloumi S. and Jaoua A., Conceptual Information Retrieval System based on cooperative conceptual data reduction. *1St International Conference on Information & Communication Technologies : from Theory to Applications*, Syria, (2004).

20. Nafkha I., Elloumi S., Jaoua A., Using Concept Formal Analysis for Cooperative Information Retrieval. *Concept Lattices and their applications Workshop* (CLA'04), VSB-TU Ostrava, September 23th-24th, 2004.

21. Rijsbergen C.J. Van, A non-classical logic for information retrieval. *The Computer Journal* 29, 1986, N 6, pages 481-485.

22. Rijsbergen C.J. Van, A new theorical framework for information retrieval. *Proceeding of the 1986-ACM Conference on Research and Development in Information Retrieval*, 1986, pages 194-200.

23. Salton G., Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. *Addison Wesley*, 1989.

24. Salton G., A. Wang and C. S. YANG, A vector space model for automatic indexing, *Communication of the ACM* 18, 1975, N°11, pages 613-620.

25. Salton G., Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science* 41, 1990, N°4, pages 288-297.

26. Salton G. and Buckley C., Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science (JASIS)*. Vol.41, N°4, pages 288-297, 1990.

27. Waller G. W. and Kraft D.H., A mathematical model of a weighted Boolean retrieval system. *Information Processing and Management* (1997), N°15, pages 235-245.