BR-Explorer: An FCA-based algorithm for Information Retrieval

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, Malika Smaïl-Tabbone

BR-Explorer: An FCA-based algorithm for Information Retrieval

Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smaïl-Tabbone

UMR 7503 LORIA, BP 239, 54506 Vandœuvre-lès-Nancy, FRANCE
{messai,devignes,napoli,smail}@loria.fr
http://www.loria.fr/∼messai

Abstract. In this paper we present BR-Explorer, an FCA-based algorithm that addresses the problem of retrieving the relevant objects for a given query. Initially, a formal context representing the relation between a set of objects and the corresponding set of attributes is given, and the associated concept lattice is built. BR-Explorer starts by generating a formal concept representing the considered query, and classifies this query concept in the concept lattice. Then, BR-Explorer tries to locate the so-called "pivot" concept in the concept lattice, for building step by step the query result (considering the pivot superconcepts in the concept lattice). Finally, BR-Explorer returns a set of objects ranked with respect to their relevance w.r.t. the query.

1 Introduction

Information Retrieval (IR) has always been a major concern in Formal Concept Analysis (FCA) [4,1]. Indeed, an obvious analogy exists between object-attribute and document-term tables. Accordingly, formal concepts of a concept lattice may be seen as a pair (*answer*, *query*) where the *query* corresponds to the intent of the concept while the *answer* corresponds to the extent of the concept. The subsumption relation between formal concepts can be considered as a specialization/generalization relation between such queries. Moreover, the way formal concepts are classified in a concept lattice allows an easy browsing (navigation) of the lattice structure and hence provides a second way for using concept lattices in IR, namely IR by navigation. The two forms of IR using concept lattices (by querying and by browsing) can easily be combined. Such a combination provides more precise results retrieved in a flexible way. In fact, a query can first be submitted to a lattice-based IR system to locate the formal concept containing the most precise answer. Once the answer concept is identified, additional results can be identified by browsing the concept lattice.

This paper details an FCA-based IR algorithm called BR-Explorer. BR-Explorer exceeds the classical document-term field to deal with a more specialized one, namely bioinformatic data bases retrieval [3,5]. This paper gives a formal description and generalization of the research work presented in [3] showing that it may be generalized to IR based on FCA principles.

2 Formal definitions

In the following, we suppose that there exists a formal context $\mathbb{K} = (G, M, I)$, where G is a set of objects, M a set of attributes, and I is an incidence relation (on $G \times M$). The set of concepts that may be built from the formal context $\mathbb{K} = (G, M, I)$ is denoted by $\mathfrak{B}(G, M, I)$, and the resulting concept lattice by $\underline{\mathfrak{B}}(G, M, I)$ [2]. Figure 1 represents an example of formal context and its corresponding concept lattice (drown with the *ConExp* [1] system). BR-Explorer tries

G \ M	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8
g_1	×	×			×			
g_2	×	×			×			
g_3		×	×		×			
g_4		×	×			×		×
g_5		×	×			×		×
g_6		×	×			×	×	×
g_7			×	×		×	×	

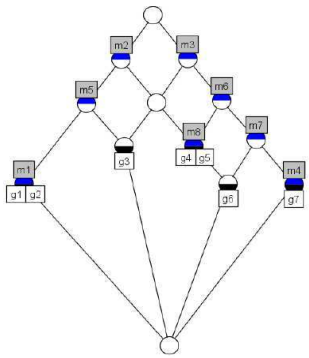

Fig. 1. The formal context $\mathbb{K} = (G, M, I)$ and its corresponding concept lattice $\underline{\mathfrak{B}}(G, M, I)$

answer a query $Q = (\{x\}, \{x\}')$ where $\{x\}'$ is a set of given attributes describing the constraints that must be satisfied by objects to be retrieved.

Definition 1 (Query). *A query Q is a pair $(\{x\}, \{x\}')$ where $\{x\}'$ is a set of attributes and x is a "dummy object" satisfying the constraints expressed by the attributes in $\{x\}'$.*

As in the well-known FCA-based IR algorithms [1], BR-Explorer retrieves objects by classifying the query in a concept lattice organizing the considered objects. The insertion of the query in the concept lattice can be considered as the addition of a new entry in the initial formal context. Consider as an example the query $Q = (\{x\}, \{x\}')$, where $\{x\}' = \{m_4, m_6, m_7\}$. The addition of this query to the formal context $\mathbb{K} = (G, M, I)$ yields the formal context $\mathbb{K}_Q = (G_Q, M_Q, I_Q)$. To allow this extension of formal context, we define the operator $\oplus$.

Definition 2 (Extension of a formal context). *For a formal context $\mathbb{K} = (G, M, I)$ and a query $Q = (\{x\}, \{x\}')$ we define the addition operator $\oplus$ as follows:*

$$(G, M, I) \oplus (\{x\}, \{x\}') = (G \cup \{x\}, M \cup \{x\}', I \cup (\{x\}, \{x\}'))$$

[1] http://sourceforge.net/projects/conexp

In this way, two alternatives are possible: computing the new concept lattice from scratch or using an incremental classification algorithm such as [6]. The second alternative has been chosen in the present research work. The concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ associated to the formal context $\mathbb{K}_Q = (G_Q, M_Q, I_Q)$ is shown in figure 2.

Before starting the retrieval of relevant objects for the considered query, two things must be defined: (1) the relevance criterion allowing to decide whether an object is relevant to the query or not and (2) the retrieval starting point in the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ allowing to avoid the whole concept lattice scan.

Definition 3 (Relevance criterion). *Consider an entry* $(\{a\}, \{a\}')$ *in a formal context* $\mathbb{K} = (G, M, I)$*, and a query* $Q = (\{x\}, \{x\}')$*. The object* a *is relevant with respect to* Q *if and only if* $\{a\}' \cap \{x\}' \neq \emptyset$*, i.e. there is at least one attribute in* $\{x\}'$ *shared with the object* a*.*

The retrieval starting point is the formal concept representing the query in the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$. Depending on whether $\{x\}$ is closed [2] in G_Q or not, this concept may be different from the query Q. In all the cases this concept is called the *pivot* concept, denoted $P$and defined as follows.

Definition 4 (Pivot concept). *Consider* $\mathbb{K} = (G, M, I)$ *a formal context and* $Q = (\{x\}, \{x\}')$ *a query. The pivot concept in the concept lattice* $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ *of the formal context* $\mathbb{K}_Q = (G_Q, M_Q, I_Q)$ *is the concept* $P = (\{x\}'', \{x\}')$*.*

In the example introduced above, the pivot concept in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ is $P = (\{g_7, x\}, \{m_4, m_6, m_7\})$ (figure 2).

Considering the relevance criterion defined above, the following proposition can be stated.

Proposition 1. *Consider a formal context* $\mathbb{K} = (G, M, I)$ *and a query* $Q = (\{x\}, \{x\}')$*. All the relevant objects with respect to* Q *in* G *are in the extent of the pivot concept* $P = (\{x\}'', \{x\}')$*, namely* $\{x\}''$*, and the extents of the pivot superconcepts in the concept lattice* $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$*.*

Proof. Consider the objects in $\{x\}''$, the extent of the pivot concept. According to the definition of the pivot concept $P = (\{x\}'', \{x\}')$ (i.e. definition 4) and the definition of relevance (i.e. definition 3), all the objects in $\{x\}''$ are relevant with respect to the query $Q = (\{x\}, \{x\}')$ since they share all the attributes in $\{x\}'$, the query intent. For the case of the pivot superconcepts, consider $C = (A, B)$ a superconcept of P in $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$, i.e. $P = (\{x\}'', \{x\}') \sqsubseteq C = (A, B)$. Then, by definition of the lattice ordering, $B \subseteq \{x\}'$, meaning that each object in A shares at least an element with $\{x\}'$, and hence is relevant.

Based on the subsumption relation, the so-called *upper cover* defined hereafter allows to scan only the interesting parts of the concept lattice for retrieving the relevant objects of the considered query.

Definition 5 (upper cover). *(1) Consider a formal context* $\mathbb{K} = (G, M, I)$*, the set of formal concepts* $\mathfrak{B}(G, M, I)$ *and the concept lattice* $\underline{\mathfrak{B}}(G, M, I)$*. The*

upper cover of a formal concept $Y \in \mathfrak{B}(G, M, I)$ *is the set of all direct upper neighbors [2] of* Y *in* $\underline{\mathfrak{B}}(G, M, I)$*:*

$$upper\text{–}cover(Y) = \{C \in \mathfrak{B}(G, M, I) \,|\, Y \sqsubseteq C \, and \, \nexists \, Z \in \mathfrak{B}(G, M, I) \,|\, Y \sqsubseteq Z \sqsubseteq C\}$$

(2) Given a set $\{C_j\}_{j \in J}$ *of formal concepts in* $\mathfrak{B}(G, M, I)$*, the upper cover of the set* $\{C_j\}_{j \in J}$ *(*J *a set of elements in* $\mathbb{N}$*) is defined as the union of the upper cover of each concept* C_j*:*

$$upper-cover(\{C_j\}_{j \in J}) = \bigcup_{j \in J} upper-cover(C_j)$$

3 The BR-Explorer algorithm

Consider a query $Q = (\{x\}, \{x\}')$, a formal context $\mathbb{K} = (G, M, I)$ and the concept lattice $\underline{\mathfrak{B}}(G, M, I)$. BR-Explorer proceeds as follows. Firstly, the query

Algorithm 1 BR-Explorer

Require: $\mathbb{K}$ = (G,M,I), $\underline{\mathfrak{B}}$(G,M,I) and Q = ({x},{x}′)
Ensure: $R_{objects}$

```
 1: Insert Q into 𝔅(G,M,I)
 2: P = ({x}″,{x}′) := Locate_Pivot(𝔅(G_Q,M_Q,I_Q),Q)
 3: n := 1 /* n is the level in 𝔅(G_Q,M_Q,I_Q) from P */
 4: SUBS_{n−1} := {P}
 5: rank := 1
 6: if {x}″ ≠ {x} then
 7:    R_rank := {x}″ \ {x}
 8:    R_objects := (rank,R_rank)
 9:    rank := rank + 1
10: end if
11: while SUBS_{n−1} ≠ ∅ do
12:    SUBS_n := upper-covers(SUBS_{n−1})
13:    R_rank := ∅
14:    for all C = (A,B) ∈ SUBS_n such that B ≠ ∅ do
15:       R_rank := R_rank ∪ A
16:    end for
17:    EmergingObjects := R_rank \ ({x} ∪ R_1,R_2,...,R_{rank−1})
18:    R_objects := R_objects ∪ (rank,EmergingObjects)
19:    n := n + 1
20:    rank := rank + 1
21: end while
```

$Q = (\{x\}, \{x\}')$ is classified and inserted in the lattice $\underline{\mathfrak{B}}(G, M, I)$ (Algorithm 1 line 1). This classification yields a new concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$ and a pivot concept $P = (\{x\}'', \{x\}')$ (line 2; P is given by the procedure *Locate_Pivot*: algorithm 2). The set of objects that are in $\{x\}''$ and in the extents of the

Algorithm 2 Locate_Pivot

Require: $\underline{\mathfrak{B}}$(G$_Q$,M$_Q$,I$_Q$) and Q = ({x},{x}$'$)
Ensure: P = ({x}$''$,{x}$'$)
1: found := false
2: SUBS := {$\bot$} /* $\bot$ is the bottom concept in $\underline{\mathfrak{B}}$(G$_Q$,M$_Q$,I$_Q$) */
3: **while** ! found **do**
4: **for each** C = (A,B) $\in$ SUBS **do**
5: **if** {x}$'$ = B **then**
6: P := C
7: found := true
8: break
9: **else if** X$'$ $\subset$ B **then**
10: SUBS := upper-cover(SUBS)
11: break
12: **end if**
13: **end for**
14: **end while**

superconcepts of P are assigned to the result set $\mathcal{R}_{objects}$ (lines 8 and 18) as a pair (*rank*, *set of objects*) (line 18). This pair is interpreted as: the objects in *set of objects* have the rank *rank* in the final result for the considered query. This form of $\mathcal{R}_{objects}$ allows the memorization of the rank of each object in the final result during the objects insertion in the result.

The result construction starts by considering the set $SUBS_0$ containing only the concept P ($SUBS_0 = \{P\}$). At this step, if $\{x\}''\backslash\{x\} \neq \emptyset$ then the objects in $\{x\}''\backslash\{x\}$ are added to $\mathcal{R}_{objects}$ with the appropriate rank (first rank in this case). The next step consists in considering $SUBS_1 = \textit{upper-cover}(SUBS_0)$. The set of objects in the extents of the concepts in $SUBS_1$ and not already in the result (the emerging objects) are added to $\mathcal{R}_{objects}$ with the corresponding rank. The algorithm proceeds in the same way for $SUBS_2$, $SUBS_3$ etc until an empty set $SUBS_n$ is reached. At each step i, if the concept $\top$ appears in the set of concepts $SUBS_i$ and if the intent of $\top$ is the empty set, then the objects in its extent are ignored. Figure 2 shows a running example of BR-Explorer where the formal context considered is $\mathbb{K} = (G, M, I)$ given in figure 1 and the query is $Q = (\{x\}, \{m_4, m_6, m_7\})$. The pivot concept returned by the procedure *Locate_Pivot* is $P = (\{g_7, x\}, \{m_4, m_6, m_7\})$ and the result is $\mathcal{R}_{Objects} = \{(1, \{g_7\}), (2, \{g_6\}), (3, \{g_4, g_5\})\}$.

The way BR-Explorer proceeds to retrieve relevant objects for a given query allows this algorithm to achieve high performances in term of *recall* and *precision*. In fact, BR-Explorer involves in part an operation considered as query refinement in [1] by looking for relevant objects in the pivot superconcepts. In this way, BR-Explorer increases the recall without decreasing the precision since as stated in proposition 1, the objects that are in the extents of the pivot superconcepts are also relevant w.r.t. the query (proposition 1).

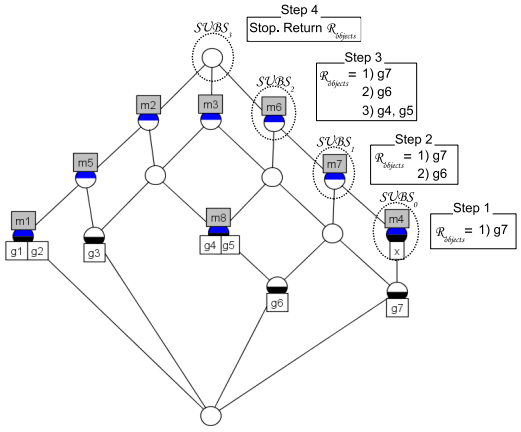

Fig. 2. Steps of the BR-Explorer execution on the concept lattice $\underline{\mathfrak{B}}(G_Q, M_Q, I_Q)$

4 Conclusion

The algorithm BR-Explorer presented in this paper is aimed at IR and query answering in a concept lattice. It has been successfully applied in biology [3] and may be used in other application domains, that can be formalized using a set of objects and a set of corresponding attributes. One original aspect characterizing BR-Explorer is the way objects are retrieved and the way the result is progressively built. This gives to BR-Explorer a different behavior, contrasting other IR approaches in the field of FCA, such as those presented in [1] and [4].

References

1. Claudio Carpineto and Giovanni Romano. *Concept Data Analysis: Theory and Applications*. John Wiley & Sons, 2004.
2. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis*. Springer, mathematical foundations edition, 1999.
3. Nizar Messai, Marie-Dominique Devignes, Amedeo Napoli, and Malika Smail-Tabbone. Querying a bioinformatic data sources registry with concept lattices. In *Proceedings of ICCS 2005, Kassel, Germany, July 18-22, 2005*, pages 323–336.
4. Uta Priss. Lattice-based Information Retrieval. *Knowledge Organization*, 27(3):132–142, 2000.
5. Malika Smail-Tabbone, Shazia Osman, Nizar Messai, Amedeo Napoli, and Marie-Dominique Devignes. Bioregistry: a structured metadata repository for bioinformatic databases. In *Proceedings of CompLife'05, Konstanz, Germany, September 25-27, 2005*.
6. Dean van der Merwe, Sergei A. Obiedkov, and Derrick G. Kourie. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In *Proceedings of ICFCA 2004, Sydney, Australia, February 23-26, 2004*, pages 372–385.