Extracting Threshold Conceptual Structures from Web Documents

Ciobanu, Gabriel; Horne, Ross; Vaideanu, Cristian

2014

Ciobanu, G., Horne, R., & Văideanu, C. (2015). Extracting Threshold Conceptual Structures from Web Documents. Lecture Notes in Computer Science, 8577, 130-144.

https://hdl.handle.net/10356/81014

https://doi.org/10.1007/978-3-319-08389-6_12

Extracting Threshold Conceptual Structures from Web Documents

Gabriel Ciobanu[1], Ross Horne[1], and Cristian Văideanu[2]

[1] Romanian Academy, Institute of Computer Science, Iaşi
[2] A.I.Cuza University of Iaşi, Faculty of Mathematics
gabriel@info.uaic.ro, {ross.horne,cvaideanu}@gmail.com

Abstract. In this paper we describe an iterative approach based on formal concept analysis to refine the information retrieval process. Based on weights for ranking documents we define a weighted formal context. We use a Galois connection to introduce a new type of formal concept that allows us to work with specific thresholds for searching words in Web documents. By increasing the threshold, we obtain smaller lattices with more relevant concepts, thus improving the retrieval of more specific items. We use techniques for processing large data sets in parallel, to generate sequences of Galois lattices, overcoming the time complexity of building a lattice for an entire large context.

1 Introduction

Formal concept analysis (FCA) is a data analysis technique based on lattice theory which provides effective methods for conceptual clustering and knowledge representation [16]. In the last decade, FCA has been used for various applications like cluster analysis, semantic Web, image processing and knowledge discovering. It has been proved useful particularly for information retrieval (IR) applications, providing a support structure which has improved search strategies like query refinement, ranking, document classification and combinations of various views for semistructured data.

In FCA-based information retrieval applications, the documents usually serve as formal objects and the index terms as formal attributes. Thus, the formal context coincides in fact with the so-called document-term matrix. The concept lattice (Galois lattice) associated can be interpreted as a search space that can be explored using different retrieval strategies. A query submitted by the user is assimilated with a concept intent, while the documents retrieved by the system are its extent. There are many approaches which have been developed methods to explore the concept lattice in order to find the relevant documents for a user's query. Thus in [1] the retrieved documents are found using the distance from the concepts of the lattice to the "query concept", i.e. the concept whose intent coincides with the query. In [11] the lattice-based IR techniques for classifying and searching relevant bioinformatic data sources is used to find the needed information exploring only the superconcepts of the query concept in the conceptual structure. In [4] it is proposed a FCA-based approach for semantic indexing and

N. Hernandez et al. (Eds.): ICCS 2014, LNAI 8577, pp. 130–144, 2014.
DOI: 10.1007/978-3-319-08389-6_12,

retrieval using the Galois lattice as a semantic index and as a search space to model terms.

Organising the data as a lattice structure has many advantages from the perspective of information retrieval: an enhanced browsing retrieval, better possibilities for extracting knowledge from the conceptual hierarchy and for identifying the conceptual associations for query reformulation. However, these approaches have one important limitation: the size of the concept lattice can be very large with respect to their underlying context. Some researchers have proposed different solutions to overcome this problem. Thus, the system CREDO generates a small portion of the lattice, typically consisting of the query concept and its neighbours [2]. In [14] several formal methods of combining a document collection with a multi-faceted thesaurus are presented. The lattice-based system FaIR improve the retrieval by partitioning the set of documents into smaller sets.

Here we address the issue of extracting conceptual structures from large collections of Web documents at a lower complexity. Thus, we propose a model which add to the lattice-based IR approach some relevance conditions based on terms weight. First of all, we use a formal context having an extended incidence relation that includes some weights associated with each term and document. Based on this context type, we generate a Galois lattice which connects terms and relevant documents. Using a relevance condition for documents which depends on a threshold value, we get a new type of formal concepts, namely threshold formal concepts (t-concepts). Our dynamical structure offers a more flexible way to extract, organise and represent the information contained in Web pages. Every time the threshold value t is increased, a new threshold Galois connection between the document set and the term set is created. Thus, we build a sequence of t-concept lattices which can iteratively refine the set of retrieved documents, as a result of the user's query. By modifying the terms weight threshold, the density of the formal context can be rapidly adjusted such that the complexity of the corresponding Galois lattice to be decreased. We also use the t-concept lattices to define a new method for ranking the documents returned by the system. We develop here a technique combining the navigation through the system of t-concept lattices built and a relevance condition based on terms weight. Firstly, the rank of a document is computed as the length of the shortest path from the query concept to the concepts whose extent contains the document. When many documents are equally relevant, we refine the ranking by using the sequence of t-Galois lattices.

As we already noticed, the complexity of the concept lattice is one of the major problems of any lattice-based IR applications. In this paper we overcome this shortcoming by using the power of parallel distributed calculus applied to our model of threshold concept lattices. The MapReduce software framework allows us to generate in parallel the concepts of every fixed t-lattice from the sequence, and then all the t-lattices in the sequence. The power of the parallel calculus across a distributed cluster of processors help us to overcome the complexity of the generating process.

The rest of the paper is organised as follows. In Section 2 we briefly overview the FCA results. Then we introduce the notion of threshold concept lattice in Section 3. In Section 4 we describe the iterative retrieval process, including some notes about the MapReduce algorithm, and the ranking method for the documents. Finally, our conclusions are presented in Section 5.

2 Basics of Formal Concept Analysis

Formal concept analysis was proposed as a mathematical method of data analysis [16]. This approach takes a binary relation between a set of objects and a set of attributes, which are properties of the objects, and then creates a space of "concepts".

Definition 1. *A formal context is a triple (X, Y, I) with X, Y being abstract sets and $I \subseteq X \times Y$ a relation between X and Y. We call the elements of X objects, those of Y attributes and I the incidence relation of the context (X, Y, I).If xIy, we say that "object x has the attribute y".*

Definition 2. *Let (X, Y, I) be a formal context. The applications $\alpha : (\mathcal{P}(X), \subseteq) \to (\mathcal{P}(Y), \subseteq)$ and $\beta : (\mathcal{P}(Y), \subseteq) \to (\mathcal{P}(X), \subseteq)$ defined by*

$$\alpha(A) = \{y \in Y \mid \forall x \in A,\ xIy\},\ A \neq \varnothing,\ \alpha(\varnothing) = Y$$
$$\beta(B) = \{x \in X \mid \forall y \in B,\ xIy\},\ B \neq \varnothing,\ \beta\ (\varnothing) = X$$

are called the derivation operators.

It is known that the derivation operators α, β form an antitone Galois connection between the ordered sets$(\mathcal{P}(X), \subseteq)$ and $(\mathcal{P}(Y), \subseteq)$ i.e. α, β are decreasing and the operators $\beta \circ \alpha$, $\alpha \circ \beta$ are extensive.

Definition 3. *Let (X, Y, I) be a formal context. A pair $(A, B) \in \mathcal{P}(X) \times \mathcal{P}(Y)$ is said to be a formal concept of (X, Y, I) if $\alpha(A) = B$ and $\beta(B) = A$.*
The sets A, B are called the extent and the intent of the formal concept (A, B), respectively. The set of all formal concepts is denoted by $\mathcal{C}(X, Y, I)$.

If (A, B) is a formal concept, then B can be interpreted as a set of terms appearing in a query, while A represents the documents retrieved. These sets are maximal according to the properties stated in the definition. On the set of formal concepts we can define a hierarchical order, denoted as $\leq$; thus, we say that the concept (A_1, B_1) is less (more specific) than the concept (A_2, B_2) or that (A_2, B_2) is greater (more general) than (A_1, B_1)) if $A_1 \subseteq A_2$ (equivalently $B_1 \supseteq B_2$). The basic theorem of FCA states that the set of formal concepts ordered by the $\leq$ relation forms a complete lattice [7].

Theorem 1. *The set $\mathcal{C}(X, Y, I)$ endowed with the relation $\leq$ is a complete lattice called concept lattice or Galois lattice, and is denoted by $\underline{\mathcal{C}}(X, Y, I)$.*

Example 1. An example of formal context and its concept lattice is presented in Figure 1. We find the concepts which belong to the concept lattice following the method presented in [7] :

$$
\begin{aligned}
c_1^1 &= (\{d_1, d_2, d_3, d_4, d_5, d_6\}, \varnothing) \text{ (the top concept)}\\
c_2^1 &= (\{d_2, d_3, d_4, d_5, d_6\}, \{ring\}), c_3^1 = (\{d_1, d_4, d_5, d_6\}, \{gold\})\\
c_4^1 &= (\{d_2, d_3, d_4\}, \{ring, algebra\}), c_5^1 = (\{d_2, d_3, d_5, d_6\}, \{ring, planet\})\\
c_6^1 &= (\{d_4, d_5, d_6\}, \{ring, gold\}), c_7^1 = (\{d_4\}, \{ring, gold, algebra\})\\
c_8^1 &= (\{d_5, d_6\}, \{ring, gold, planet\}), c_9^1 = (\{d_2, d_3\}, \{ring, algebra, planet\})\\
c_{10}^1 &= (\varnothing, \{ring, gold, algebra, planet\}) \text{ (the bottom concept)}
\end{aligned}
$$

	ring	*gold*	*algebra*	*planet*
d_1	0	1	0	0
d_2	1	0	1	1
d_3	1	0	1	1
d_4	1	1	1	0
d_5	1	1	0	1
d_6	1	1	0	1

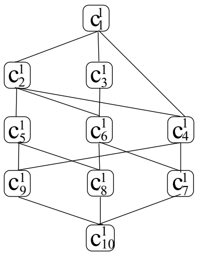

Fig. 1. A Formal Context and its Concept Lattice

In [1], based on the subconcept-superconcept relation $\leq$, a so-called neighbour relation is defined. Thus, if c_1, c_2 are two concepts in the context (X, Y, I), we say that c_1 is the nearest neighbour of c_2 if either $c_1 < c_2$ and $c_1 \leq c < c_2$ implies $c = c_1$ or $c_2 < c_1$ and $c_2 < c \leq c_1$ implies $c = c_1$. It is clear that the nearest neighbour represents the minimal refinement or enlargement of a concept in a given formal context.

3 Threshold Formal Concepts

In what follows, we consider X to be a set of documents (Web pages) and Y a set of terms (descriptors). For example, for each document we can take as attributes some selected indexed words obtained by eliminating all stopwords and very common terms, stemming words to their roots or limiting them to nouns and few descriptive adjectives and verbs [9].

We define now our model. Firstly, a new type of derivation operators, called threshold-derivation operators, are described. Let $y \in Y$ be a term. A function $w_y : X \to [0, \infty)$ is called *weight* of the term y. The weight $w_y(x)$ for $x \in X$ can be, for example, the normalised frequency of the term y in the document x, but we consider $w_y(x)$ to be the term frequency-inverse document frequency of the term y (see [10]). We consider the weighted incidence relation $\tilde{I} : X \times Y \longrightarrow [0, \infty)$, $\tilde{I}(x, y) = w_y(x)$, for all $x \in X$, $y \in Y$ and the weighted formal context $(X, Y, \tilde{I})$.

Definition 4. *Let $t \in [0, \infty)$. We define the function $\psi_t : X \to \mathcal{P}(Y)$ by*

$$\psi_t (x) = \{y \in Y | w_y (x) \geq t\}, \forall x \in X.$$

The set $\psi_t (x)$ represents all terms in Y such that their weight in the document x is greater than the threshold t. We now define a new type of derivation operators, namely the threshold-derivation operators.

Definition 5. *Let $t \in [0, \infty)$. $\alpha_t : \mathcal{P}(X) \to \mathcal{P}(Y)$ and $\beta_t : \mathcal{P}(Y) \to \mathcal{P}(X)$*

$$\begin{aligned} &\alpha_t (A) = \bigcap_{x \in A} \psi_t (x), \ \forall A \subseteq X, A \neq \varnothing, \ \alpha_t (\varnothing) = Y \text{ and} \\ &\beta_t (B) = \{x \in X \mid B \subseteq \psi_t (x)\}, \forall B \subseteq Y \end{aligned}$$

are called threshold-derivation operators (t-derivation operators).

In fact, $\alpha_t (A)$ is the set of terms which belong to all documents in A with weights in each of these documents greater than the threshold t, while $\beta_t (B)$ represents the set of documents such that the weight of any term from B is greater than t. The t-derivation operators generalise the operators defined in the previous paragraph. If we take $0 < t \leq \min\{w_y (x) \mid (x, y) \in X \times Y\}$, then we obtain $\alpha_t (A) = \alpha (A)$, for all $A \subseteq X$ and $\beta_t (B) = \beta (B)$ for all $B \subseteq Y$.

Example 2. Let $X = \{d_1, d_2, .., d_6\}$ be a set of documents, $Y = \{ring, gold, algebra, planet\}$ a set of attributes, and $\tilde{I}$ the incidence relation given in Fig.2:

	ring	*gold*	*algebra*	*planet*
d_1	0	3	0	0
d_2	4	0	3	1
d_3	3	0	2	4
d_4	1	1	2	0
d_5	4	2	0	3
d_6	2	4	0	2

Fig. 2. A Weighted Formal Context

We have

$$\begin{aligned} \alpha_{1.5} (\{d_2, d_3\}) = \psi_{1.5} (d_2) \cap \psi_{1.5} (d_3) = \\ \{ring, algebra\} \cap \{ring, algebra, planet\} = \{ring, algebra\}, \end{aligned}$$

which means that the attributes "$ring, algebra$" belong to documents d_2, d_3 and their weights are greater than the threshold $t_2 = 1.5$. We have $\alpha_{1.5} (\{d_2, d_3\}) \neq \alpha (\{d_2, d_3\}) = \{ring, algebra, planet\}$, which proves that α_t generalises α, the derivation operator defined in Example 1. If $t_3 = 2.5$, then we obtain

$$\begin{aligned} \alpha_{2.5} (\{d_2, d_3\}) = \psi_{2.5} (d_2) \cap \psi_{2.5} (d_3) = \\ \{ring, algebra\} \cap \{ring, planet\} = \{ring\}. \end{aligned}$$

Because $\beta_{1.5}\left(\{ring, algebra\}\right) = \{d_2, d_3\}$, it results that d_2, d_3 represent all documents which have "$ring, algebra$" as attributes with weights greater than 1.5. We also notice that

$$\beta_{1.5}\left(\{ring, algebra\}\right) \neq \beta\left(\{ring, algebra\}\right) = \{d_2, d_3, d_4\}.$$

To define our model of formal concepts, the following result is fundamental.

Proposition 1. *The pair (α_t, β_t) is an antitone Galois connection between the ordered sets $(\mathcal{P}(X), \subseteq)$, $(\mathcal{P}(Y), \subseteq)$.*

Proof. (i) We first prove that α_t is decreasing (i.e., $A_1 \subseteq A_2$ implies $\alpha_t(A_2) \subseteq \alpha_t(A_1)$). We have $\alpha_t(A_2) = \bigcap\limits_{x\in A_2} \psi_t(x) \subseteq \bigcap\limits_{x\in A_1} \psi_t(x) = \alpha_t(A_1)$.

(ii) Next, we show that the operator $\beta_t \circ \alpha_t$ is extensive (i.e., $A \subseteq \beta_t(\alpha_t(A))$). We have $\beta_t(\alpha_t(A)) = \beta_t\left(\bigcap\limits_{x\in A} \psi_t(x)\right) = \left\{x' \in X | \bigcap\limits_{x\in A} \psi_t(x) \subseteq \psi_t(x')\right\}$. If $x \in A$, it follows that $\bigcap\limits_{x\in A} \psi_t(x) \subseteq \psi_t(x)$, hence $x \in \beta_t(\alpha_t(A))$.

(iii) We prove that the t-operator β_t is decreasing (i.e. $B_1 \subseteq B_2$ implies $\beta_t(B_2) \subseteq \beta_t(B_1)$). Let $B_1 \subseteq B_2$. If $x \in \beta_t(B_2)$, then $B_2 \subseteq \psi_t(x)$, hence $B_1 \subseteq \psi_t(x)$.

(iv) Finally, we show that the operator $\alpha_t \circ \beta_t$ is extensive (i.e. $B \subseteq \alpha_t(\beta_t(B))$. The inclusion $B \subseteq \psi_t(x)$ is true for all $x \in \beta_t(B)$, and consequently,
$$B \subseteq \bigcap_{x\in\beta_t(B)} \psi_t(x) = \alpha_t(\beta_t(B)).$$

We study how the operators α_t, β_t depend on the threshold t.

Proposition 2. *Let $t \in [0, \infty)$, $A \subseteq X$ be a subset of documents and $B \subseteq Y$ a subset of terms. Then α_t, β_t are decreasing with respect to t, i.e.*
$$\alpha_{t+1}(A) \subseteq \alpha_t(A) \text{ and } \beta_{t+1}(B) \subseteq \beta_t(B), \forall t \in [0, \infty).$$

Proof. In the case $A \neq \varnothing$ we have:
$$\psi_{t+1}(x) = \{y \in Y \mid w_y(x) \geq t+1\} \subseteq \{y \in Y \mid w_y(x) \geq t\} = \psi_t(x),$$
for all $x \in X$, which implies that $\bigcap\limits_{x\in A} \psi_{t+1}(x) \subseteq \bigcap\limits_{x\in A} \psi_t(x)$. If $A = \varnothing$ the relation is obvious. The second inclusion results in a similar way.

We define the notion of formal concept which allows to develop more efficient methods for information retrieval.

Definition 6. *Let $t \in [0, \infty)$. A pair $(A, B) \in \mathcal{P}(X) \times \mathcal{P}(Y)$ is said to be a threshold-formal concept (t-formal concept) if*
$$\alpha_t(A) = B \text{ and } \beta_t(B) = A.$$

The set A is called threshold-extent, and the set B threshold-intent of the t-formal concept (A, B); we denote $A = ext((A, B))$, and $B = int((A, B))$. The set of t-formal concepts is denoted by $\mathcal{C}_t(X, Y, \tilde{I})$. For each document $x \in X$, we define the associate document concept $(\beta_t(\alpha_t(x)), \alpha_t(x))$, and for each word $y \in Y$, the associate term concept $(\beta_t(y), \alpha_t(\beta_t(y)))$.

Example 3. We consider the context from Example 2. We find that

$$\alpha_{1.5}\left(\{d_2, d_3\}\right) = \{ring, algebra\} \text{ and } \beta_{1.5}\left(\{ring, algebra\}\right) = \{d_2, d_3\},$$

hence the pair $(\{d_2, d_3\}, \{ring, algebra\})$ is a t-formal concept corresponding to $t_2 = 1.5$. This concept is different from the "classical " formal concept $(\{d_2, d_3\}, \{ring, algebra, planet\})$ (see Figure 1).

We now study how the t-derivation operators act on t-formal concepts.

Proposition 3. *Let $t \in [0, \infty)$, J an index set and $(A_j, B_j) \in \mathcal{C}_t(X, Y, \tilde{I})$, for all $j \in J$. Then, we have:*

(i) $\alpha_t(\bigcup_{j \in J} A_j) = \bigcap_{j \in J} \alpha_t\left(A_j\right)$
(every intersection of intents is an intent);

(ii) $\beta_t(\bigcup_{j \in J} B_j) = \bigcap_{j \in J} \beta_t\left(B_j\right)$
(every intersection of extents is an extent).

Proof. (i)

$$\alpha_t(\bigcup_{j \in J} A_j) = \bigcap_{x \in \cup A_j} \psi_t\left(x\right) = \bigcap_{j \in J} \bigcap_{x \in A_j} \psi_t\left(x\right) = \bigcap_{j \in J} \alpha_t\left(A_j\right).$$

(ii) We have

$$\beta_t(\bigcup_{j \in J} B_j) = \{x \in X \mid \bigcup_{j \in J} B_j \subseteq \psi_t\left(x\right)\} = \bigcap_{j \in J} \{x \in X \mid B_j \subseteq \psi_t\left(x\right)\}$$

$$= \bigcap_{j \in J} \beta_t\left(B_j\right).$$

Corollary 1. *(i) Every extent of a t-formal concept is the intersection of some terms extents.*
(ii) Every intent of a t-formal concept is the intersection of some documents intents.

The hierarchical order in the t-Galois lattices can be introduced in a similar way as for the Galois lattices.

Definition 7. *Let $t \in [0, \infty)$ and (A_1, B_1), $(A_2, B_2) \in \mathcal{C}_t(X,\ Y, \tilde{I})$. Then (A_1, B_1) is a subconcept of (A_2, B_2) whenever $A_1 \subseteq A_2$, and we denote this by $(A_1, B_1) \leq (A_2, B_2)$. The set of all formal concepts ordered by $\leq$ is denoted by $\underline{\mathcal{C}}_t(X, Y, \tilde{I})$.*

We prove the basic theorem on threshold-concept lattices:

Theorem 2. *Let $t \in [0, \infty)$. Then the poset $\underline{\mathcal{C}}_t(X, Y, \tilde{I})$ of t-formal concepts is a complete lattice.*

Proof. The proof is similar to the classical one. If J is an index set and $\{(A_j, B_j)|j \in J\} \subseteq \mathcal{C}_t(X, Y, \tilde{I})$ is a subset of t-formal concepts, we define the t-infimum and, respectively, the t-supremum of this set by:

$$\bigwedge_{j\in J}(A_j, B_j) = (\bigcap_{j\in J} A_j, \alpha_t(\beta_t(\bigcup_{j\in J} B_j)))$$
$$\bigvee_{j\in J}(A_j, B_j) = (\beta_t(\alpha_t(\bigcup_{j\in J} A_j)), \bigcap_{j\in J} B_j).$$

We first prove the formula for the infimum. Since $A_j = \beta_t\,(B_j)$ for all $j \in J$, and according to Proposition 3, we have

$$(\bigcap_{j\in J} A_j, \alpha_t(\beta_t(\bigcup_{j\in J} B_j))) = (\beta_t(\bigcup_{j\in J} B_j), \alpha_t(\beta_t(\bigcup_{j\in J} B_j))),$$

which means that this pair is a t-formal concept. Since $\inf\{A_j \mid j \in J\} = \bigcap_{j\in J} A_j$, and taking into account the order relation for t-concepts, it follows that the infimum of the subset $\{(A_j, B_j)|j \in J\}$ is $\bigwedge_{j\in J}(A_j, B_j)$.
The proof for the supremum formula is similar.

It is worth noting that, though the t-concepts set is a complete lattice (as in the classical case), it is qualitatively different. The t-concepts are built using a relevance condition over the terms; this dependence can be exploited to find better ways to search through a collection of Web documents.

4 Iterative Retrieval Process

Using a sequence of t-concept lattices and techniques based on querying and navigating through the corresponding lattices, we present an iterative procedure which can improve the the process of extracting information from data sets. At each step, using a relevance condition, we build a new Galois lattice through the MapReduce programming model. Thus, we obtain a decreasing sequence of concepts $\left(c^i\right)$ such that the intents of c^i are the successive queries Q_i submitted to the system, while the extents represent the retrieved documents.

Let $t_1 = 0$ be the initial threshold, $K_1 = (X, Y, \tilde{I})$ a given weighted formal context, and Q_1 the initial query. Let us describe the step i of the procedure $(i \in \mathbb{N}^*)$. Let Q_{i-1} be the subset of terms which represents the user query, obtained in the step $(i-1)$ and $t_i > t_{i-1}$, the new threshold value. Next, using the MapReduce model we build the concept lattice $\mathcal{C}_i(X, Y, \tilde{I})$, $i \in \mathbb{N}^*$ associated to the new context K_i. There are a lot of algorithms to find the Galois lattice, for example Ganter's "Next Closure Algorithm" [7], or the algorithm proposed in [8]. As inserting or deleting documents or terms in the formal context is an usual operation, it is clear that we need an incremental algorithm to build the

t-conceptual structure. Therefore, we think that the AddIntent algorithm [12] is a good choice, in order to generate the t-Galois lattice.

In what follows we describe how we used the MapReduce software framework to implement our model. First, using SPARQL, a Python script harvests strings describing resources on the Web. The Python script also tokenizes the strings, removes stop words and reduces words to stems using the platform NLTK for language processing. The data is then fed into a MapReduce pipeline. The pipeline first calculates the term frequency-inverse document frequency for the collection of documents. It filters out the top n terms for each document. Finally, it calculates the intents of all t-formal concepts. To achieve this task, it just takes two passes: the first pass gathers all the intents and extents of individual objects and attributes, and the second pass calculates the t-formal concepts. 1. First pass:

```
map phase (object, attribute) : (uri, string)
  emit (object, attribute) and (attribute, object)
reduce phase (x, ys) : (a, [b])
  pipe into second pass
```

2. Second pass:

```
map phase (x, ys) : (a, [b])
  for each y in ys
    emit (y, ys)
reduce phase (y, yss) : (a, [[a]])
  store (y, intersection yss).
```

Let us consider an example. Let $\{(a,t),(b,t),(b,u),(c,u)\}$ be the incidence relation. In one very efficient pass, we can obtain the extents and intents: $\{(a,t),(b,t,u),(c,u)\},\{(t,a,b),(u,b,c)\}$. From this we obtain the document concepts and the term concepts: $\{(t,t),(u,u)\},\{(a,a,b),(b,b),(c,b,c)\}$. The massively parallel nature of MapReduce with the grouping of keys between the map and reduce phase overcomes the complexity problem.

The sequence of Galois lattices obtained allows gradual refinement or enlargement of a query. Let us describe how we can construct the new query Q_{i+1}. When the query Q_i is submitted by the user, there are two possibilities: either the Galois lattice $\mathcal{C}_i(X,Y,\tilde{I})$ contains a concept $c^i = \left(ext\left(c^i\right), int\left(c^i\right)\right)$ such that the query can be interpreted as the intent of c^i ($int\left(c^i\right) = Q_i$), or this property is not satisfied. In the later case, following the ideas from [1], we augment the Galois lattice $\mathcal{C}_i(X,Y,\tilde{I})$ with a pseudo-concept denoted pc^i such that the intent of pc^i equals the query Q_i and as extent, we consider a set which contains one pseudodocument pd^i. We can navigate through $\mathcal{C}_i(X,Y,\tilde{I})$ starting from the concept c^i or from the pseudoconcept pc^i, and choose another concept of the lattice, say $\overline{c}^i$. Then we take $Q_{i+1} = int\left(\overline{c}^i\right)$. The concept $\overline{c}^i$ is usually chosen such that to give a minimal refinement or enlargement of c^i so that, in fact, $\overline{c}^i$ is one of the neighbours nodes (parents or children) of c^i or pc^i. One can also take $Q_{i+1} = Q_i$. It is worth noting that if c^i is a concept in $\mathcal{C}_i(X,Y,\tilde{I})$, it generally does not result that c^i is a concept in $\mathcal{C}_{i+1}(X,Y,\tilde{I})$; neither the converse property is generally true. We emphasise that, in order to traverse the conceptual hierarchy starting from the query concept, we use the breadth-first search algorithm implemented again on a parallel architecture.

We argue now that the query Q_i can be reformulated at each step such that the sequence of the retrieved extents $\left(ext\left(c^i\right)\right)_{i\in\{1,2,..,p\}}$, $p \in \mathbb{N}^*$, which depends on the threshold t_i, to decrease. At step i we must take into account three situations. In the first one we consider $Q_i = Q_{i+1}$. In the second one, we choose $\overline{c}^i$ to be a child of the concept c^i or of the pseudoconcept pc^i (we enlarge the query) and we have $ext\left(c^i\right) \supseteq ext\left(c^{i+1}\right)$, which means that we refine the information retrieved.

In the third situation, $\overline{c}^i$ is a parent of c^i or pc^i (the query is refined), hence we have $ext\left(c^i\right) \subseteq ext\left(c^{i+1}\right)$ which is the minimal enlargement of the document set.

For example, if at each step i we are in the first case, the sequence of sets $\left(ext\left(c^i\right)\right)_{i\in\{1,2,..,p\}}$ (the extents of concepts c^i) is decreasing

$$ext\left(c^{i-1}\right) \supseteq ext\left(c^i\right), \forall i \in \{1, 2, .., p\},$$

and the procedure stops when $ext\left(c^i\right)$ become $\varnothing$.

In the general case, the sequence of retrieved documents can be split into a finite number of decreasing "subsequences":

$$\begin{gathered} ext(c^1) \supseteq ext\left(c^2\right) \supseteq ... \supseteq ext\left(c^{i_1}\right), \\ ext(c^{i_1+1}) \supseteq ext\left(c^{i_1+2}\right) \supseteq ... \supseteq ext\left(c^{i_2}\right), \\ ext(c^{i_2+1}) \supseteq ext\left(c^{i_2+2}\right) \supseteq ... \supseteq ext\left(c^{i_3}\right)\dots \end{gathered}$$

such that at steps i_1, i_2, .. the user chooses the parent node associated to the query concept.

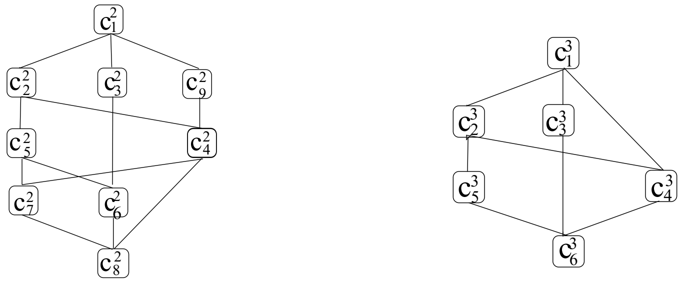

Fig. 3. Threshold $t_2 = 1.5$

Example 4. We illustrate our procedure on a small example. Let $K_1 = (X, Y, \tilde{I})$ be the formal context given in the table presented in Example 2, and the initial threshold $t_1 = 0$. The t-concept lattice $\underline{\mathcal{C}}_1\left(X, Y, \tilde{I}\right)$ is represented in Figure 1. At step 1, we choose the threshold t_2 to be 1.5.

In Figure 3, the diagram of the concept lattice $\underline{\mathcal{C}}_2(X, Y, \tilde{I})$ is sketched; the concepts are: $c_1^2 = (\{d_1, d_2, d_3, d_4, d_5, d_6\}, \varnothing)$, $c_2^2 = (\{d_2, d_3, d_5, d_6\}, \{ring\})$, $c_3^2 = (\{d_1, d_5, d_6\}, \{gold\})$, $c_4^2 = (\{d_2, d_3\}, \{ring, algebra\})$, $c_5^2 = (\{d_3, d_5, d_6\}, \{ring, planet\})$, $c_6^2 = (\{d_5, d_6\}, \{ring, gold, planet\})$, $c_7^2 = (\{d_3\},$

$\{ring, algebra,\ planet\})$, $c_8^2 = (\varnothing, \{ring, gold,\ algebra, planet\})$ and $c_9^2 = (\{d_2, d_3, d_4\}, \{algebra\})$. Next, we consider $t_3 = 2.5$ and $t_4 = 3.5$, respectively. The Hasse diagrams of the suitable Galois lattices $\underline{\mathcal{C}}_i(X, Y, \tilde{I})$, $i \in \{3, 4\}$ can be found in a similar way, and they are depicted in Figure 3 and Figure 4, respectively.

The concepts for $t_3 = 2.5$ and $t_4 = 3.5$ are $c_1^3 = (\{d_1, d_2, d_3, d_4, d_5, d_6\}, \varnothing)$, $c_2^3 = (\{d_2, d_3, d_5\}, \{ring\})$, $c_3^3 = (\{d_1, d_6\}, \{gold\})$, $c_4^3 = (\{d_2\}, \{ring, algebra\})$, $c_5^3 = (\{d_3, d_5\}, \{ring, planet\})$, $c_6^3 = (\varnothing, \{ring, gold, algebra, planet\})$, respectively $c_1^4 = (\{d_1, d_2, d_3, d_4, d_5, d_6\}, \varnothing)$, $c_2^4 = (\{d_2, d_5\}, \{ring\})$, $c_3^4 = (\{d_6\}, \{gold\})$, $c_4^4 = (\{d_3\}, \{planet\})$ and $c_5^4 = (\varnothing, \{ring, gold,\ algebra, planet\})$.

Let us see the procedure at work. One of the simplest sequences of queries is obtained, for example, by choosing $Q_1 = \{gold\}$ and take $Q_i = Q_{i+1}$, $i \in \{1, 2, 3\}$; in this case we must mention that at every step there exists a concept c^i such that $int\left(c^i\right) = Q_i, i \in \{1, 2, 3, 4\}$. The set of documents retrieved at each search, represented by $ext\left(c^i\right)$, $i \in \{1, 2, 3, 4\}$, is decreasing: $c_3^1 = \{d_1, d_4, d_5, d_6\}$, $c_3^2 = \{d_1, d_5, d_6\}$, $c_3^3 = \{d_1, d_6\}$, $c_3^4 = \{d_6\}$, $\varnothing$. We notice again that as t_i increases the documents corresponding to the query become more relevant. For example, the more relevant documents for the query $\{gold\}$ are d_6, d_1, d_5, d_4 (in this order).

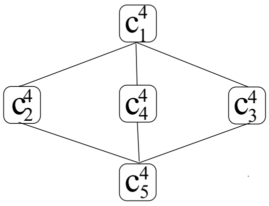

Fig. 4. Threshold $t_4 = 3.5$

Let us analyse a more complicated example. If we start with the query $Q_1 = \{ring, algebra\}$, we obtain in $\underline{\mathcal{C}}_1$ the formal concept $c_4^1 = (\{d_2, d_3, d_4\}, \{ring, algebra\})$. Since in the lattice $\underline{\mathcal{C}}_2$, there is no concept such that its intent equals Q_1, we must add to $\underline{\mathcal{C}}_2$ a pseudoconcept $pc^2 = (pd^2, \{ring, algebra\})$ and choose Q_2 to be the intent of the nearest neighbour concept of pc^2. Thus, if we choose $Q_2 = \{ring, algebra, planet\}$ the extent $ext(c_7^2) = \{d_3\}$ of the obtained concept decreases. If $Q_2 = \{ring\}$ we get the concept $c_2^2{=}(\{d_2, d_3, d_5, d_6\}, \{ring\})$, and so the set of retrieved documents $ext(c_2^2) = \{d_2, d_3, d_5, d_6\}$ does not decrease. Thus, the user either can find new relevant documents (in our case d_5, d_6), or can deduce documents d_2, d_4 to be less important than d_3. For the next step of the procedure we take $Q_3 = \{ring\}$, and find in the lattice $\underline{\mathcal{C}}_3$ the extent $ext(c_2^3) = \{d_2, d_3, d_5\}$, which means that the sequence of retrieved documents decreases again. Finally, in step 4, we can take $Q_4 = \{ring\}$ and we obtain $ext\left(c_3^4\right) = \{d_2, d_5\})$. Hence, at each step, the user has the possibility to choose the query such that the sequence $\left(ext\left(c^i\right)\right)_{i \in \{1,2,3,4\}}$ decreases.

4.1 Documents Ranking

It is obvious that for large collections, the set of documents which is associated with the query could be very big. Using the hierarchical structure of the t-Galois lattices, we propose a method to rank the documents retrieved by the system.

The FCA-based IR model defined in [15] implements an explicit relevance feedback. When the query concept has too many parent or child concepts, it is difficult for the user to inspect all the intents of these concepts to reformulate the query. To fix this problem, the authors define an order relation on the set of these children (or parents) nodes, a so-called *preference* relation. By means of this relation, the user can select the most relevant documents. This model is different from that developed in [13], where the choice of the relevant documents which are used to the query expansion is made directly by the user. In [1] the query is merged into the concepts document space and the similarity between the query and a document is computed as the length of the shortest path linking the concept query with the concept whose extent equals the set of attributes of the document.

To compute the rank of a document, we approach a slightly different method combined with the use of the t-concept lattices. Let $p \in \mathbb{N}^*$ and $F = \{t_1, t_2, ..., t_p\} \subset [0, \infty)\,,\ t_1 < t_2 < ... < t_p$ be the set of values of the threshold (the optimal choice for the set F depends on the document collection and is established through experiments). For each $t_i \in F$, $i \in \{1, 2, .., p\}$, we denote with $\underline{\mathcal{C}}_i$, $\mathcal{C}_i$ the Galois lattice and the set of concepts that belong to $\underline{\mathcal{C}}_i$, respectively. We remind that the distance between two concepts C_i^1, $C_i^2 \in \mathcal{C}_i$ is defined by:

$$d_i : \mathcal{C}_i \times \mathcal{C}_i \to [0, \infty)\,,\ d_i\left(C_i^1, C_i^2\right) = \text{length of the shortest path from } C_i^1 \text{ to } C_i^2.$$

Now, let $t_i \in F$ be a fixed threshold, $Q \subset Y$ the query, and $d \in X$ a document. We define the similarity between d and Q in the threshold lattice $\underline{\mathcal{C}}_i$ as being the least of the distances from the query concept C_Q to the nodes C_i^d which contain d and are superconcepts of C_Q:

$$sim_i : X \to [0, \infty)\,,\ sim_i\,(d, Q) = \min\{d_i\left(C_i^d, C_Q\right) \mid C_i^d \in \underline{\mathcal{C}}_i^d,\ C_i^d \supseteq C_Q\}.$$

In fact this similarity is given by the number of minimal refinements to modify the query such that to equal the intent of a concept C_i^d. It is natural to consider a document that better matches the query, to have a greater rank.

Definition 8. *A document d_1 is ranked ahead of d_2 in the lattice $\underline{\mathcal{C}}_i$, related to a query Q, if $sim_i\,(d_1, Q) < sim_i\,(d_2, Q)$.*

Example 5. In example 4, if we set $Q = \{ring, algebra\}$ the document d_3 is ranked better than d_5 in the lattice $\underline{\mathcal{C}}_1$ (see Figure 1) because $sim_1\,(d_3, Q) = 0$ and $sim_i\,(d_5, Q) = 1$.

Taking into account the sequence of t-Galois lattices and using the method previously described, we order the documents in the concept lattice $\underline{\mathcal{C}}_1$. The set of documents which have the same rank still could be very large. We now augment the defined rank method with a criterion based on terms' weight in documents. Let d_1 and d_2 be two documents of equal rank, related to Q. We use the iterative process described in previous section. Let $\left(c^i\right)_{i\in\{1,2,..,p\}}$ be a sequence of concepts obtained by applying the above mentioned procedure, $c^i = \left(ext\left(c^i\right), int\left(c^i\right)\right)$, $ext\left(c^i\right) \supseteq ext\left(c^{i+1}\right)$. Since when i increases the concepts c^i become more relevant, it is natural to define:

Definition 9. *Let d_1, $d_2 \in X$ with $sim_1\left(d_1, Q\right) = sim_1\left(d_2, Q\right)$. The document d_1 is ranked ahead d_2 related to the query Q and we denote by $d_1 \succ d_2$, if there is $i \in \{1, 2, .., p\}$ such that $d_1 \in ext\left(c^i\right)$ and $d_2 \notin ext\left(c^i\right)$.*

Example 6. Using Example 4, we take $Q = \{ring, algebra\}$. The documents d_2, d_3, d_4 have, in lattice $\underline{\mathcal{C}}_1$, the rank 0, while d_1, d_5, d_6 the rank 1. We choose the concepts c_4^1 and c_4^3, which intents equal the query, $ext\left(c_4^1\right) = \{d_2, d_3, d_4\} \supseteq ext\left(c_4^3\right) = \{d_2\}$. Because c_4^3 is more relevant than c_4^1 and $d_2 \in ext\left(c_4^3\right)$, $d_3, d_4 \notin ext\left(c_4^3\right)$ it results that d_2 is more relevant than d_3 and d_4. Now, we consider the sequence $\left(c_2^i\right)_{i\in\{1,2,3,4\}}$, $c_2^i \in \mathcal{C}_i$, for all $i \in \{1, 2, 3, 4\}$ of parent concepts of c_Q. The documents that belong to the most relevant concepts have the bigger rank. We have $\{d_2, .., d_6\} \supseteq \{d_2, d_3, d_5, d_6\} \supseteq \{d_2, d_3, d_5\} \supseteq \{d_2, d_5\}$, hence we find that $d_3 \succ d_4$ and $d_5 \succ d_6 \succ d_1$.

Once the top ranked documents are returned in response to the user's request Q, the system can add to the query the most relevant terms selected from these documents, thus improving the retrieval process.

5 Conclusion

In this paper we developed a theoretical model which integrates some relevance conditions to an IR model based on concept lattices. Using the weights of the terms, we described a new type of concepts, namely threshold formal concepts, which offers a more efficient way to extract the information from a collection of Web documents. The sequence of hierarchical structures we built provides a dynamical IR model. During the search process, as the threshold increases the t-formal concepts become more relevant, thus obtaining a more rapid access to the documents needed. Depending on the feedback, the user can navigate, not only in the same t-concept lattice, but in any lattice in the sequence and dynamically refine or enlarge the query. Lattices generated with FCA techniques can be complex, thus difficult to use in practical applications. Due to the weighted form of the context, even if we work with big data tables, by increasing the threshold t we obtain a sequence of t-formal contexts with a lower density which can decrease the complexity of the method.

The novelty of our approach comes also from the use of a parallel and distributed system to implement the model. We used the MapReduce software framework which allows to achieve the NLP filtering, and to generate the sequence of the t-concept lattices. The preliminary results are very promising. Using parallel computation, we overcome the complexity of the iterative process, even in the case of large and dense t-formal contexts.

We have also introduced a new ranking method which combines hierarchical navigation in a t-Galois lattice with relevance conditions. Thus, output documents are ranked according to their similarity with the query based on the structure of the t-lattices, and using the fact that documents which occur in more relevant concepts are more relevant. As future work, we intend to improve our experimental studies over large, dense contexts. Thus, we would like to study what values of the threshold t should the user choose in order to optimise the described iterative process. By experimenting with a wide range of values, we hope to observe which features of the weighted formal context yield a fast retrieval behaviour.

Acknowledgements. We thank the anonymous reviewers for their comments. The work was supported by a grant of the Romanian Authority for Scientific Research, project number PN-II-ID-PCE-2011-3-0919.

References

1. Carpineto, C., Romano, G.: Order-Theoretical Ranking. Journal of the American Society for Information Science 51, 587–601 (2000)
2. Carpineto, C., Romano, G.: Exploiting the Potential of Concept Lattices for Information Retrieval with CREDO. J. Univ. Comput. Sci. 10, 985–1013 (2004)
3. Cigarrán, J.M., Gonzalo, J., Peñas, A., Verdejo, F.: Browsing Search Results via Formal Concept Analysis: Automatic Selection of Attributes. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 74–87. Springer, Heidelberg (2004)
4. Codocedo, V., Lykourentzou, I., Napoli, A.: A Contribution to Semantic Indexing and Retrieval Based on FCA; An Application to Song Datasets. In: Proceedings CLA. CEUR Workshop, vol. 972, pp. 257–268 (2012)
5. Dau, F., Ducrou, J., Eklund, P.: Concept Similarity and Related Categories in SearchSleuth. In: Eklund, P., Haemmerlé, O. (eds.) ICCS 2008. LNCS (LNAI), vol. 5113, pp. 255–268. Springer, Heidelberg (2008)
6. El Qadi, A., Aboutajdin, D., Ennouary, Y.: Formal Concept Analysis for Information Retrieval. Int'l Journal of Computer Science and Information Security 7, 119–125 (2010)
7. Ganter, B., Wille, R.: Formal Concept Analysis. Mathematical Foundations. Springer (1999)
8. Godin, R., Missaoui, R., Alaoui, H.: Incremental Concept Formation Algorithms Based on Galois Lattices. Computational Intelligence 11, 246–267 (1995)
9. Karp, D., Schabes, Y., Zaidel, M., Egedi, D.: A Freely Available Wide Coverage Morphological Analyzer for English. In: Proceedings 14th COLING, pp. 950–955 (1992)
10. Manning, C.D., Raghavan, P., Schütze, H.: An Introduction to Information Retrieval. Cambridge University Press (2009)
11. Messai, N., Devignes, M.-D., Napoli, A., Smaïl-Tabbone, M.: Querying a Bioinformatic Data Sources Registry with Concept Lattices. In: Dau, F., Mugnier, M.-L., Stumme, G. (eds.) ICCS 2005. LNCS (LNAI), vol. 3596, pp. 323–336. Springer, Heidelberg (2005)

12. van der Merwe, D., Obiedkov, S., Kourie, D.: AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. In: Eklund, P. (ed.) ICFCA 2004. LNCS (LNAI), vol. 2961, pp. 372–385. Springer, Heidelberg (2004)
13. Nauer, E., Toussaint, Y.: Dynamical Modification of Context for an Iterative and Interactive Information Retrieval Process on the Web. In: Proceedings CLA. CEUR Workshop, vol. 331, 12 p. (2007)
14. Priss, U.: Lattice-Based Information Retrieval. Knowledge Organization 27, 132–142 (2000)
15. Spyratos, N., Meghini, C.: Preference-Based Query Tuning Through Refinement/Enlargement in a Formal Context. In: Dix, J., Hegner, S.J. (eds.) FoIKS 2006. LNCS, vol. 3861, pp. 278–293. Springer, Heidelberg (2006)
16. Wille, R.: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel (1982)