Using pattern structures to support information retrieval with Formal Concept Analysis

Victor Codocedo, Ioanna Lykourentzou, Hernan Astudillo, Amedeo Napoli

Using pattern structures to support information retrieval with Formal Concept Analysis

Víctor Codocedo[1⋆], Ioanna Lykourentzou[1,2⋆⋆], Hernán Astudillo[3], and Amedeo Napoli[1]

[1] LORIA - CNRS - INRIA - Univesité de Lorraine, BP 239, 54506 Vandœuvre-les-Nancy.
victor.codocedo@loria.fr, amedeo.napoli@loria.fr,
[2] Centre de Recherche Public Henri Tudor - 29, avenue John F. Kennedy L-1855 Luxembourg-Kirchberg, Luxembourg
ioanna.lykourentzou@tudor.lu
[3] Universidad Técnica Federico Santa María - Avenida España 1680 - Valparaíso, Chile
hernan@inf.utfsm.cl

Abstract. In this paper we introduce a novel approach to information retrieval (IR) based on Formal Concept Analysis (FCA). The use of concept lattices to support the task of document retrieval in IR has proven effective since they allow querying in the space of terms modelled by concept intents and navigation in the space of documents modelled by concept extents. However, current approaches use binary representations to illustrate the relations between documents and terms ("document D contains term T") and disregard useful information present in document corpora ("document D contains X references to term T"). We propose using pattern structures, an extension of FCA on multi-valued and numerical data, to address the above. Given a set of weighted document-term relations, a concept lattice based on pattern structures is built and explored to find documents satisfying a given user query. We present the meaning and capabilities of this approach, as well as results of its application over a classic IR document corpus.

Keywords: Formal Concept Analysis, Interval Pattern Mining, Information Retrieval

1 Introduction

Information retrieval (IR), is a problem of lasting interest for the research community. Among the tasks comprising the IR domain, document retrieval (i.e. the search and ranking of documents that are relevant to an original user query from a given document corpus) is one of the most popular in the field given its importance in everyday routines. In the wide spectrum of techniques applied to support document retrieval, formal concept analysis (FCA) has gained interest in the last years [3–6, 16] because of its robust framework and the qualities of a concept lattice.

Formal concept analysis (FCA) is a mathematical formalism used for data analysis and classification, which relies on the dualistic understanding of concepts as consisting

⋆ Part of the Quaero Programme, funded by OSEO, French State agency for innovation.
⋆⋆ Supported by the National Research Fund, Luxembourg, and cofunded under the Marie Curie Actions of the European Commission (FP7-COFUND).

of an extent (the objects that belong to the concept) and an intent (the attributes that those objects share) organized in a lattice structure called a concept lattice [7]. We refer to FCA-based information retrieval as CL4IR which stands for"concept lattices for information retrieval".

In a typical CL4IR approach, a binary table of documents and terms is created and then, using FCA algorithms, the respective concept lattice is created. This lattice contains several *formal concepts*, each defined by a set of documents (extent) and the set of terms that they share (intent). Thus, the lattice provides a multiple hierarchical classification of documents and terms which can be navigated and searched as an index, to retrieve concepts that are "close" or "similar" to the original query concept. In this way a CL4IR system exploits the connections of concepts within the lattice to find relevant documents for a given query and takes advantage of the lattice structure to enrich the answer in different ways: by navigating the lattice to look for approximate answers through generalizations and specifications of the original query concept's intent [2, 14], by enriching the term vocabulary with a thesaurus [5] or by directly integrating external knowledge sources [16]. Nevertheless, current CL4IR systems are restricted by the binary nature of their data (a document can either contain a given term or not). Consequently, they can work only with Boolean-like queries, which is an important limitation w.r.t. other IR approaches such as vector-space ranking methods [13] that allow partial-matching documents to be considered as possible answers.

In this article we present a novel CL4IR approach, which deals with numerical datasets, i.e. document-term relations, where a document is annotated by a term with a certain weight. This approach provides CL4IR systems with an extended *query space*, on which vector-space ranking methods can be adapted and applied. In parallel, this approach retains the main advantage of using lattices as the document search index, which is the provision and exploration potential of the *complete query space*. Our approach is based on the pattern structures framework, an extension of FCA to deal with complex data [7]. Given a numerical table representing weighted associations between documents and terms, we apply pattern structures to build the extended query space, while we also introduce steps for reducing and simplifying the document search within the constructed query space. We illustrate our approach through running example on a classical IR dataset and by comparing our results, in terms of precision and recall, to those reported in the literature. Furthermore we provide a discussion on the meaning and capabilities of the proposed approach. The remainder of the paper is organized as follows. Section 2 provides an introduction to the use of formal concept analysis for document retrieval. Section 3 describes the pattern structure framework and details the proposed CL4IR approach which can be applied on numerical datasets. Section 4 presents the experiments. Finally, Section 5 concludes the paper.

2 Concept Lattices for Information Retrieval

The setting of a typical concept lattice for information retrieval (CL4IR) application is given by a formal context $\mathcal{K} = (D, T, I)$ made of a set of documents D, set of terms T and an incidence relation $I = \{(d_i, t_j)\}$ indicating that document d_i contains

term t_j. Table 1 illustrates a document-term formal context created from a corpus of 9 documents and 12 terms.

	human	interface	computer	user	system	response	time	EPS	survey	tree	graph	minor
d_1	x	x	x									
d_2			x	x	x	x	x		x			
d_3		x		x	x			x				
d_4	x				x			x				
d_5				x		x	x					
d_6										x		
d_7										x	x	
d_8										x	x	x
d_9									x		x	x
q *										x	x	

* Grey row represents the *query*.

Table 1: A term-document formal context including the query q.

Given a user query $q = \{t_1, t_2...t_{|q|}\}$, the document retrieval task consists in returning a set of documents ordered by "relevance" w.r.t. the query q. In CL4IR systems a query can be represented as a virtual document containing the set of terms $\{t_1, t_2...t_{|q|}\}$. Then, the query is inserted in the formal context as another object and the incidence relation set I is updated to include the relations of the virtual query-document and its terms. The formal context becomes $\mathcal{K}_q = (D + \{q\}, T, I + \{(q, t_i)_{i..|q|}\})$.

The standard procedure to find "relevant" documents within the concept lattice consists in identifying the *query concept* (which is defined as the *object concept* of the virtual object q and denoted by $\gamma(q) = ((q')', q')$) and concepts related to the *query concept* (for example, its superconcepts) which can provide further results. We refer to the later concepts as "answer concepts". For the formal context in Table 1, consider the query q with terms "graph" and "tree" (grey row). Figure 1 shows the concept lattice derived from this formal context (including the query). The *query concept* corresponds to concept 17 and contains in its extent documents d_7 and d_8 which satisfy the query and can be retrieved to the user. The superconcepts of the *query concept* (concepts 7 and 8) contain documents d_6 and d_9 which can also be retrieved. Different relevance measures can be used to rank the retrieved documents. For example, the topological distance within the lattice between the *query concept* and the "answer concepts" (i.e. concepts partially satisfying the query) can be calculated and in this case documents $d7$ and $d8$ are at distance 0 (more relevant), while d_6 and d_9 are at distance 1 (less relevant). Other such measures include semantic distance, extent intersection, and Jaccard similarity [2, 15, 5].

More generally, the concept lattice defines a *query space* where each formal concept C can be considered as a conjunctive Boolean query (i.e. a query where the constraint

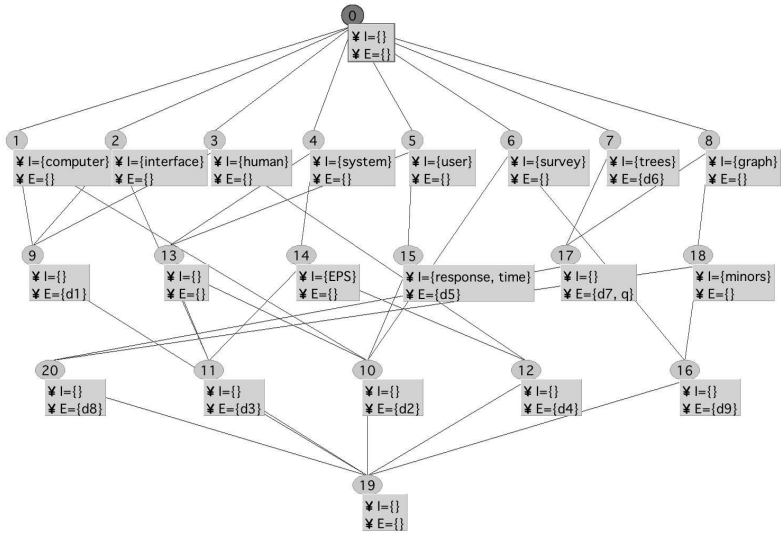

Fig. 1: Concept lattice in reduced notation derived from a document-term formal context including the query.

is given by the conjunction of the attributes in the intent of C) and a combination of formal concepts provides disjunction and negation (e.g. The union of concepts 7 and 8 in Figure 1 satisfies the disjunctive query "graph" or "tree"). Unfortunately, the binary case is the "ideal world". In most real-world datasets the relation between a document and a term is built w.r.t. a measure such as frequency, distance or weight involving a range of numerical values [13].

A document corpus can be defined as a term-document matrix $A = [a_{ij}]$, where terms $t_i \in T$ are in rows, documents $d_j \in D$ in columns and each cell a_{ij} of the matrix represents the "value" of the term t_i in the document d_j, given by a function $val(d_j, t_i)$ (weight, frequency, etc.). In order to work with this kind of datasets, a CL4IR system can resort to interordinal scaling [8] by simply assigning an incidence relation when a term in a document has a value within a given range, i.e. $I = \{(d, t)|val(d, t) > 0\}$). However, interordinal scaling could greatly increase the complexity for IR tasks [12] as it induces redundancy as shown in [11]. To the best of the authors' knowledge, a CL4IR system directly dealing with a weighted term-document dataset is not yet reported in the FCA nor in IR literature. In the following, we present a method and an implementation of a CL4IR approach dealing with numerical datasets.

3 CL4IR with many-valued datasets

3.1 Pattern structure framework

Here, we introduce the pattern structure framework firstly described in [7].

A pattern structure $\mathcal{K} = (G, (P, \sqcap), \delta)$ is a generalization of a formal context. In $\mathcal{K}$, G is a set of objects, $(P, \sqcap)$ is a semi-lattice of object descriptions and $\delta : G \to P$ is a

mapping associating a description to an object. The description of an object $g \in G$ is a vector of intervals $v = \langle [l_i, r_i] \rangle_{i \in \{1..|M|\}}$, where $v \in P$, $l_i, r_i \in \mathbb{R}$ and $l_i \leq r_i$.

In $(P, \sqcap)$ the *similarity* operator $\sqcap$ applied to $v_a = \langle [l_i^1, r_i^1] \rangle$ and $v_b = \langle [l_i^2, r_i^2] \rangle$ yields the convex hull $v_a \sqcap v_b = \langle [min(l_i^1, l_i^2), max(r_i^1, r_i^2)] \rangle$ where $i \in \{1..|M|\}$. The associated subsumption relation is defined as $v_a \sqcap v_b = v_a \iff v_a \sqsubseteq v_b$.

A Galois connection between $\wp(G)$ (powerset of G) and $(P, \sqcap)$ is defined as follows:

$$X^{\square} = \bigsqcap_{g \in X} \delta(g) \; ; \; v^{\square} = \{g \in G | v \sqsubseteq \delta(g)\}$$

where $X^{\square}$ represents the common description to all objects in X while $v^{\square}$ represents the set of objects respecting the description v. A pair (X, v) such as $X^{\square} = v$ and $v^{\square} = X$ is called a *interval pattern concept (ip-concept)* with extent X and pattern intent v. Ip-concepts can be ordered in an interval pattern concept lattice (ip-concept lattice). Algorithms for computing ip-concepts from an interval pattern structure are proposed in [11, 7].

3.2 CL4IR based on pattern structures

A document corpus or a term-document matrix, as described at the end of Section 2, can naturally be represented as a many-valued context [8] $\mathcal{K} = (D, T, W, I)$, where $W = \{val(d_j, t_i)\}_{\forall d_j \in D, t_i \in T}$ and $I = (d_j, t_i, w_k); f(d_j, t_i) = w_k, w_k \in W$. Table 2 shows an example containing 9 documents (white rows) and 12 terms. The value in a cell represents the "relative frequency" of a term in a document, i.e. the ratio between the amount of times a term appears in a document and the total amount of terms occurrences in the document. Like in the binary case, a query $q = \{t_1, t_2, ..t_{|q|}\}$ is considered as a virtual document and included in the many-valued context which becomes $\mathcal{K}^q = (D + \{q\}, T, W, I + \{val(q, t_i)\}_{\forall t_i \in q})$. The cells of the query contain also a "relative frequency" value. The query $q = \{$*"graph", "tree"*$\}$ is illustrated in the grey row in Table 2 (e.g. $val(q, graph) = 1/2 = 0.5$).

To deal with $\mathcal{K}^q$, we define the pattern structure as $(D + \{q\}, (P, \sqcap), \delta)$ where interval patterns in P contain the interval-vector representation of documents in $|T|$ dimensions (one for each term). The mapping $\delta(d) = \langle [val(d, t_i), val(d, t_i)]_{i \in [1..|T|]} \rangle$ assigns an interval pattern representation to a document (or the virtual query-document) consisting of a zero-length interval for each term existing in T at the value of the term in the document (e.g. in Table 2, $\delta(d_1) = \langle [0.33, 0.33][0.33, 0.33][0.33, 0.33][0, 0] \dots [0, 0] \rangle$, where $[0, 0]$ is represented by $[-]$). The similarity operator $\sqcap$ applied to two interval patterns returns the convex hull between their document representations. From the pattern structure we construct the ip-concept lattice representing the *query space* which will be used to retrieve documents in a similar way as binary approaches.

The *query concept* is still considered as the *object concept* of q. However the semantic of the *query space* changes. While in the binary case the *query space* represents a pool of Boolean query possibilities, here the *query space* can be considered as a vector space where the query is grouped with documents having similar representations. For example, consider the first three columns in Table 3 where each row represents an ip-concept. Concept 1 is the *query concept* which in its extent includes documents d_7

and its interval pattern (intent) only includes zero-length intervals in all 12 dimensions, making the description of the query identical to the description of d_7. Concept 2 is a superconcept of 1, whose extent contains d_7 and d_8. This time, there are only 9 zero-length intervals in all 12 dimensions. Concept 2 is less similar to the query than concept 1 w.r.t 3 dimensions. Following with concept 3, we can see that the later is less similar to the query than concept 1 w.r.t. 4 dimensions. We get in this way a "natural" ranking of the concepts.

In order to rank ip-concepts we rely on the notion of *maximal distance* within an interval pattern. For illustrating this notion, we will use the geometrical interpretation of patterns already introduced in [11]. Let us consider the 2-dimensional case with two ip-concepts in Figure 2, namely $Z_1 = (\{q, A, B, C\}, \langle[2,7][2,7]\rangle)$ (clear rectangle) and $Z_2 = (\{q, A, B, C, D\}, \langle[1,7][1,7]\rangle)$ (dark rectangle). For ranking an ip-concept Z_i w.r.t. the query, we will consider the "maximal distance" possible between any two objects in the extent of Z_i, which in the case of Z_1 is between objects q and C and for Z_2 is between q and D. Thus, this distance is actually the Euclidean distance between the edges of the interval vector. Table 3 presents the retrieved ip-concepts for the query in Table 2 ranked by *maximal distance* in column 4.

	human	interface	computer	user	system	response	time	EPS	survey	tree	graph	minor
d_1	0.33	0.33	0.33	0	0	0	0	0	0	0	0	0
d_2	0	0	0.16	0.16	0.16	0.16	0.16	0	0.16	0	0	0
d_3	0	0.25	0	0.25	0.25	0	0	0.25	0	0	0	0
d_4	0.25	0	0	0	0.5	0	0	0.25	0	0	0	0
d_5	0	0	0	0.33	0	0.33	0.33	0	0	0	0	0
d_6	0	0	0	0	0	0	0	0	0	1	0	0
d_7	0	0	0	0	0	0	0	0	0	0.5	0.5	0
d_8	0	0	0	0	0	0	0	0	0	0.33	0.33	0.33
d_9	0	0	0	0	0	0	0	0	0.33	0	0.33	0.33
q [a]	0	0	0	0	0	0	0	0	0	0.5	0.5	0

Table 2: Many-valued document term context (including query).

[a] Grey row represents the *query concept.*

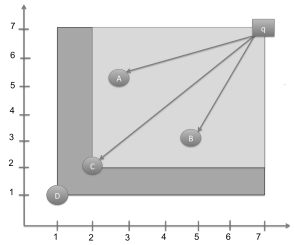

Fig. 2: Two interval patterns in the *query space*.

Id	Extent	Pattern intent	max_dist
1	q, d_7 *	$\langle[-][-][-][-][-][-][-][-][-][-][0.5,0.5][0.5,0.5][-]\rangle$	0
2	q, d_7, d_8	$\langle[-][-][-][-][-][-][-][-][-][-][0.33,0.5][0.33,0.5][0,0.33]\rangle$	0.408
3	q, d_7, d_8, d_9	$\langle[-][-][-][-][-][-][-][-][-][0,0.33][0,0.5][0.33,0.5][0,0.33]\rangle$	0.704
4	q, d_6, d_7	$\langle[-][-][-][-][-][-][-][-][-][0.5,1][0,0.5][-]\rangle$	0.707
5	q, d_2, d_7	$\langle[-][-][0,0.16][0,0.16][0,0.16][0,0.16][0,0.16][-][0,0.16][0,0.5][0,0.5][-]\rangle$	0.808
6	q, d_3, d_7	$\langle[-][0,0.25][-][0,0.25][0,0.25][-][-][0,0.25][-][0,0.5][0,0.5][-]\rangle$	0.866
7	q, d_1, d_7	$\langle[0,0.33][0,0.33][0,0.33][-][-][-][-][-][-][0,0.5][0,0.5][-]\rangle$	0.909
8	q, d_5, d_7	$\langle[-][-][-][0,0.33][-][0,0.33][0,0.33][-][-][0,0.5][0,0.5][-]\rangle$	0.909
9	q, d_4, d_7	$\langle[0,0.25][-][-][-][0,0.5][-][-][0,0.25][-][0,0.5][0,0.5][-]\rangle$	0.935

* Grey row represents the *query concept.*

Table 3: Extents and Intents of concepts in Figure 2 presenting the cosine similarity between its edges ($[-]$ represents the zero-length interval $[0,0]$).

3.3 Dealing with real-world datasets

Calculating a concept lattice is an expensive task which can yield a large amount of concepts making it prohibitive for large document corpora. The scenario is worst for pattern structures since for every concept the size of the intent is set to the whole set of attributes adding even more complexity. Calculating the whole *query space* of a term-document matrix is not advisable, since for a given query only a small part of the whole space is required. In order to avoid a sizeable *query space* in each step of the retrieval process, progressive actions to filter data are performed. In the following, we describe the retrieval process and each action.

1. Constructing the pattern structure: The process starts with the input of a query $q = \{t_1, t_2, ..., t_{|q|}\}$ and ends after the pattern structure containing the virtual query-document is created. We include in the set of documents only those which contain at least a given number of the terms provided in the query, which can be performed at a negligible cost by firstly storing documents and terms in a relational database. The set of terms only include those provided in the query. The minimum number of terms for a document is left as a parameter of the process.

2. Constructing the ip-concept lattice: This step receives the pattern structure in order to create an ip-concept lattice. A standard FCA algorithm, namely Ganter's algorithm [8] is used for this purpose. However the algorithm has been adapted for the present task.

Many ip-concepts found in the interval pattern lattice are not useful for document retrieval purposes. For example, the framework creates ip-concepts with documents which do not share terms (e.g. consider the interval $\langle[0,1][0,1][0,1]\rangle$ created from the documents sharing no terms with orthogonal representations $v_1 = \langle[0,0][1,1][1,1]\rangle$ and $v_2 = \langle[1,1][0,0][0,0]\rangle$). We denominate these concepts *non-informational*.

In order to reduce the amount of *non-informational* concepts, we modified the $\sqcap$ operator in the set of ordered patterns $(P, \sqcap)$ such as $[l,r] \sqcap [0,0] = [*]$ and $[l,r] \sqcap [*] = [*]; \forall l, r \in \mathbb{R}$. The interval $[*]$ has been used before to indicate absence of similarity [10]. Let $Z_i = (X_i, v_i)$ be an ip-concept, then $\rho(Z_i)$ represents the number of intervals different from $[*]$ in v_i. We call $\rho(Z_i)$ the dimensionality of Z_i. For a second ip-concept $Z_j = (X_j, v_j)$ is easy to show that $(Z_i \leq Z_j \iff v_j \sqsubseteq v_i) \implies \rho(Z_i) \leq \rho(Z_j)$. We use a threshold of minimal dimensionality (min_dim) to reduce the amount of ip-concepts calculated. Consider this analogous to the use of a minimal support in the construction of an iceberg lattice [18].

4 Experiments and Discussion

To test the validity of the proposed approach, we applied it on a popular IR dataset which is openly available. We refer to this implementation as "ip-CL4IR". The CISI dataset[4] consists of 1460 documents and 35 queries, each one containing a set of valid answers. Documents contain text in natural language and queries are given as a set of terms connected by Boolean operators. In our experiments, we converted documents to collections of weighted terms and stored them in a relational database. The weighting

[4] `http://ftp.cs.cornell.edu/pub/smart/cisi/`

measure used was term frequency-inverse document frequency (*tf.idf*) [13]. Boolean operators in the query were ignored since they do not provide meaning in the vector-space model (except in the extended Boolean model case [17] not considered in this work). The virtual query-document was constructed using the inverse document frequencies calculated from the dataset for each of its terms.

After receiving a query, ip-CL4IR consults the database and extracts all documents that contain at least 2 terms of the query (as described in Section 3.3, this value is a parameter of ip-CL4IR). The ip-concept lattice is computed using a minimal dimensionality of $min\_dim = 2$, to keep consistency w.r.t. the restriction given for the creation of the pattern structure. The *query concept* is searched in the lattice and its superconcepts are retrieved and ranked using the Euclidean distance between the boundaries of their interval patterns. Cosine distance (instead of Euclidean distance) was also calculated showing better results. Table 4 shows the results for 11-point precision of fixed recall and 6 measures of precision for the top 5,10 and 20 ranked documents retrieved. Results on an implementation based on concept lattice-based ranking (CLR) [2] using the same dataset and a simple binnarization of the relation document-term is reported along with our results for comparison purposes.

	ip-CL4IR	CLR	EM
11-point IAP[a]	**0.232**	0.191	0.174
MAP[b]	**0.202**	0.163	0.145
Precision@5	0.257	0.206	**0.285**
Precision@10	0.251	0.174	**0.257**
Precision@20	**0.245**	0.174	0.207
Recall@5	0.032	0.049	**0.057**
Recall@10	0.060	0.073	**0.079**
Recall@20	**0.146**	0.112	0.123

Table 4: CISI dataset. Results for 35 queries.

[a] Interpolated average precision
[b] Mean average precision

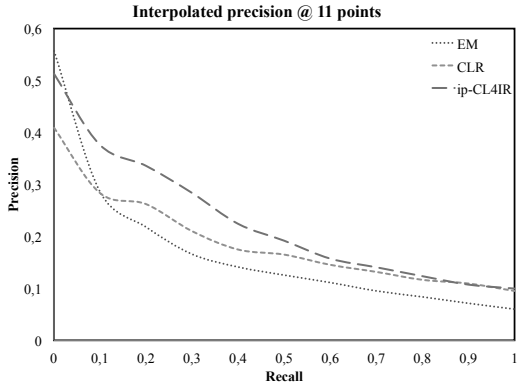

Fig. 3: Interpolated precision in 11 points of recall

Table 4 reports the results in 8 measures for ip-CL4IR, a reported CL4IR system called concept lattice-based ranking (CLR) [2] and a naive approach called exact matching (EM) where documents are ranked according to how many terms have in common w.r.t. the query. The second row contains the values of the interpolated average precision (IAP) over 11-points of recall illustrated on Figure 3. Interpolated precision in a given recall point r_i in Figure 3 indicates the best precision value in the interval $[r_i, r_{i+1}[$. From Figure 3, the interpolated precision in the recall point 0 for ip-CL4IR is the best precision obtained in the recall interval $[0, 1[$ equal to 0.51. The third row contains the values of the mean average precision (MAP) calculated over the precision values for each valid document found in the ranked documents retrieved by a system for each query. For example, given query if the first valid document is found in the third position of the ranking it has a precision value of 0.3. If the second is found in the fifth position its precision is 0.2 and the MAP is 0.25. IAP and MAP are standard information retrieval measures [13] to evaluate ranked results from a retrieval system. The remaining

rows present values of precision and recall in the first 5 (*@5*), 10 (*@10*) and 20 (*@20*) ranked documents from each system. Boldface entries indicate the best values for the three systems.

Values in Table 4 show a better performance of ip-CL4IR on 4 of the 8 measures while EM is better in the remaining 4, namely precision and recall in the first 5 and 10 ranked documents. This indicates that EM is actually better to recognize documents very close to the query, but for documents with less elements in common with the query, EM is not very precise. This can be better appreciated in Figure 3 where the interpolated precision values of ip-CL4IR quickly overcome those of EM which is only better in 1 of the 11 recall points. This fact is also supported by the significant difference in the values of IAP and MAP between ip-CL4IR and EM. For the 35 queries in the dataset, our approach took 42.23 seconds (1.2 seconds per query) to execute while for CLR took 1550.333 (44.29 seconds per query) showing an impressive enhancement in the computational time required to retrieve documents, a key issue in document retrieval. Both these times include lattice construction. Using better measures which consider the correlation among terms, or including external knowledge sources like term taxonomies may improve greatly the quality of the answers provided by our approach. These issues are currently planned as future work. These experiments were performed in an Intel Xeon machine running at 2.27 GHz with 62 GB of RAM memory.

There are many perspectives for our approach, however the most important is the full exploitation of the ip-lattice structure to improve the quality in the answers. While our principal goal in this article is to describe a general process to directly support numeric term-document datasets in a concept lattice-based information retrieval system, we argue that different IR tasks (some already supported on CL4IR systems) can be also supported on ip-CL4IR for example, document clustering [1], user feedback inclusion [6] and recommendation [9].

5 Conclusions

In this article we introduce a CL4IR approach which is able to deal directly with numerical datasets through the use of the pattern structure framework (ip-CL4IR). We provide a method and a process to construct an interval pattern concept lattice (ip-concept lattice) which can be used as a document index. We present the idea of an ip-concept lattice as a *query space* which can be navigated in order to find relevant documents. We also provide means to rank these documents using vector-based distances. The feasibility of our approach is validated through its application on a popular IR dataset for which we present precision and recall values contrasted to those reported in the literature showing a better performance in the overall list of ranked documents and an impressive enhancement in the time needed to answer a single query.

The perspectives for our approach are numerous, ranging from the improvement of its answers, its application on different real-world datasets, but most importantly, the full exploitation of the lattice structure to support different IR tasks.

References

1. C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of Web clustering engines. *ACM Computing Surveys*, 41(3):1–38, July 2009.
2. C. Carpineto and G. Romano. Order theoretical ranking. *Journal of the American Society for Information Science*, 51(7):587–601, 2000.
3. C. Carpineto and G. Romano. Exploiting the potential of concept lattices for information retrieval with CREDO. *Journal of Universal Computer Science*, 10:985 – 1013, 2004.
4. C. Carpineto and G. Romano. Using Concept Lattices for Text Retrieval and Mining. *Formal Concept Analysis*, pages 161–179, Jan. 2005.
5. V. Codocedo, I. Lykourentzou, and A. Napoli. Semantic querying of data guided by Formal Concept Analysis. In *Formal Concept Analysis for Artificial Intelligence Workshop at ECAI 2012*, 2012.
6. S. Ferré. Camelis: a logical information system to organise and browse a collection of documents. *International Journal of General Systems*, 38(4):379–403, 2009.
7. B. Ganter and S. O. Kuznetsov. Pattern Structures and their projections. *Conceptual Structures: Broadening the Base*, 2001.
8. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Dec. 1999.
9. D. I. Ignatov and S. O. Kuznetsov. Concept-based Recommendations for Internet Advertisement. *CoRR*, abs/0906.4, 2009.
10. M. Kaytoue, Z. Assaghir, A. Napoli, and S. O. Kuznetsov. Embedding tolerance relations in formal concept analysis. In *Proceedings of the 19th ACM international conference on Information and knowledge management - CIKM '10*, page 1689, New York, New York, USA, Oct. 2010. ACM Press.
11. M. Kaytoue, S. O. Kuznetsov, and A. Napoli. Revisiting numerical pattern mining with formal concept analysis. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence - Volume Volume Two*, pages 1342–1347, Nov. 2011.
12. S. O. Kuznetsov. Pattern Structures for Analyzing Complex Data. In *Proceedings of the 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing*, volume 5908 of *Lecture Notes in Computer Science*, pages 33–44. Springer Berlin Heidelberg, Dec. 2009.
13. C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press (Online edition), 1 edition, 2009.
14. N. Messai, M.-D. Devignes, A. Napoli, and M. Smaïl-Tabbone. Querying a bioinformatic data sources registry with concept lattices. In *Proceedings of the 13th international conference on Conceptual Structures: common Semantics for Sharing Knowledge*, volume 3596 of *Lecture Notes in Computer Science*, July 2005.
15. N. Messai, M.-D. Devignes, A. Napoli, and M. Smaïl-Tabbone. Using Domain Knowledge to Guide Lattice-based Complex Data Exploration. In *Proceedings of the 2010 conference on ECAI 2010: 19th European Conference on Artificial Intelligence*, pages 847–852, 2010.
16. U. Priss. Lattice-based Information Retrieval. *Knowledge Organization*, 27:132 – 142, 2000.
17. G. Salton, E. A. Fox, and H. Wu. Extended Boolean information retrieval. *Communications of the ACM*, 26(11):1022–1036, Nov. 1983.
18. G. Stumme, R. Taouil, Y. Bastide, and L. Lakhal. Conceptual clustering with iceberg concept lattices. In *Proc. GI-Fachgruppentreffen Maschinelles Lernen (FGML'01)*, Universität Dortmund 763, October 2001.