VERSITA

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **12**, No 1

Sofia • 2012

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2012-0003

Analysis of a Vector Space Model, Latent Semantic Indexing and Formal Concept Analysis for Information Retrieval

Ch. Aswani Kumar[1], *M. Radvansky*[2], *J. Annapurna*[3]

[1]*School of Information Technology and Engineering, VIT University, Vellore, India*
[2]*VSB Technical University of Ostrava, Ostrava, Czech Republic*
[3]*School of Computing Science and Engineering, VIT University, Vellore, India*
Email: cherukuri@acm.org

Abstract: *Latent Semantic Indexing (LSI), a variant of classical Vector Space Model (VSM), is an Information Retrieval (IR) model that attempts to capture the latent semantic relationship between the data items. Mathematical lattices, under the framework of Formal Concept Analysis (FCA), represent conceptual hierarchies in data and retrieve the information. However, both LSI and FCA use the data represented in the form of matrices. The objective of this paper is to systematically analyze VSM, LSI and FCA for the task of IR using standard and real life datasets.*

Keywords: *Formal concept analysis, Information Retrieval, latent semantic indexing, vector space model.*

1. Introduction

Information Retrieval (IR) deals with the representation, storage, organization and access to information items. IR is highly iterative in human interactive process aimed at retrieving the documents that are related to users' information needs. The human interaction consists in submitting information needed as a query, analyzing the ranked and retrieved documents, modifying the query and submitting iteratively until the actual documents related to the need are found or the search process is terminated by the user. Several techniques including a simple keyword to advanced NLP are available for developing IR systems. Various available IR models include a Boolean model, Vector Space Model (VSM), Probabilistic Model, Language

Model [18]. Several variants of these classical models are also available in the references concerning different retrieval tasks [35].

In VSM each document in the collection is a list of the main terms and their frequency in the document is counted. In VSM a document is regarded as a vector of terms. Each unique term in the document collection corresponds to a dimension in the space and a value indicates the frequency of this term. A user query is also considered as a document in the search space. Both the document and the query vectors provide the locations of the objects in the term-document space. The query vector is compared with the document vector for finding similarity.

Latent Semantic Indexing (LSI) extends classical VSM by modeling the term-document relationship using reduced dimension representation computed by the matrix rank reduction technique of Singular Value Decomposition (SVD) [12, 16]. This reduction results in the approximation of the original data matrix as semantic space, which reflects the major associative patterns in the data while ignoring the noise caused by word usage. Subsequently the queries are projected and processed in the lower dimensional space to find similarities with the documents.

Conceptual IR systems aim to address the limitations of the classical keyword systems and identify the conceptual associations and links between the documents. Emerging from the order and lattice theory, Formal Concept Analysis (FCA) analyzes the data which describe the relationship between the set of objects and the set of attributes of a particular domain [22]. From such description of the data in the form of formal context, FCA produces a hierarchically ordered conceptual structure called a concept lattice and a collection of attribute implications. In IR applications the document sets can be arranged as a formal context and the documents contents can be described by a set of concepts [14]. The organization of such formal concepts in the form of a lattice structure provides the user with easy navigation of the search space reducing the cognitive overload.

As far as we know, no analysis is available in literature studying these three conceptual models. This paper aims to address them with systematic analysis. Section 2 provides a brief background of VSM, LSI and FCA. We also discuss IR using FCA. Experimental results are presented in Section 3 and analysis is presented in Section 4.

2. Background

This section provides a glimpse on VSM, LSI, FCA models. We also provide a brief background on FCA for IR tasks.

2.1. Vector space model

The basic premise of VSM is that the meaning of a document can be derived from the terms constituting the document. VSM represents both documents and queries as vectors in a high dimensional space in which each dimension of the space corresponds to a term in the document collection [35]. The vectors from the collection of the documents can be collected as a matrix called term-document matrix. The retrieval performance of VSM model depends on the term weighting,

which indicates the degree of relationship between a term and a document. The references indicate that indexing with a term weight is more efficient than the binary systems. The cosine of the angle between the document and the query vector is used as numeric similarity between the vectors.

2.2. Latent semantic indexing

LSI is a variant of VSM, aimed at addressing the problem of synonymy and polysemy that plague the classical vector models. However, unlike VSM in which each term or attribute in the dataset is considered as a dimension in the feature space, LSI approximates the source space with fewer dimensions. To accomplish this decomposition of the original space, LSI uses matrix algebra technique termed SVD. For a data matrix of $\boldsymbol{A}$ of size with t rows and d columns with rank r, SVD finds the low rank approximation to $\boldsymbol{A}$ called $\boldsymbol{A}_k$ as

$$\boldsymbol{A}_k = \boldsymbol{U}_k \boldsymbol{S}_k \boldsymbol{V}_k^{\mathrm{T}}$$

where $\boldsymbol{U}_k$ is the $t \times k$ term-concept matrix, $\boldsymbol{S}_k$ is the $k \times k$ term-concept matrix and $\boldsymbol{V}_k^{\mathrm{T}}$ is the $k \times d$ concept-document matrix. This decomposition is under the assumption of orthogonality of the features in the original space. The mathematical advantage of SVD is that the obtained low rank matrix $\boldsymbol{A}_k$ is the best possible least-squares fit to $\boldsymbol{A}$. Through the process of removing $r–k$ dimensions through SVD, LSI uses only significant dimensions for further analysis or retrieval, removing the synonymy and polysemy effects. Some references have discussed several theoretical models of LSI and practical applications which have provided better understanding [12, 20, 23, 33 and the references therein].

2.3. Formal concept analysis

FCA is a mathematical framework under the branch of lattice theory [22]. FCA provides tools to identify meaningful groupings of objects that share common attributes and analyze hierarchies of these groupings. The basic notion of FCA is formal context ($\boldsymbol{G}$, $\boldsymbol{M}$, I) where $\boldsymbol{G}$ is a set of objects, $\boldsymbol{M}$ is a set of attributes and I is the relation between $\boldsymbol{G}$ and $\boldsymbol{M}$. From formal context, a formal concept can be defined as an ordered pair ($\boldsymbol{A}$, $\boldsymbol{B}$) where $\boldsymbol{A} \subseteq \boldsymbol{G}$ and $\boldsymbol{B} \subseteq \boldsymbol{M}$ are known as extent and intent of the concept. The set of these formal concepts forms a hierarchical and partially ordered structure called a concept lattice. These formal concepts constitute the nodes of the structure. Unlike a node in trees which has only one parent, a node in lattice can have multiple parents through many-many relationship. This concept lattice is a mathematically complete lattice, i.e., for each subset of concepts, the greatest common subconcept and the least common super concept exist. Generally the formal concepts are considered as clusters hidden in the data. Since the extent $\boldsymbol{A}$ contains a set of objects sharing common attributes given in $\boldsymbol{B}$, similarly the intent $\boldsymbol{B}$ consists of those attributes that commonly belong to the objects given in $\boldsymbol{A}$. By navigating upward or downward, one can easily find the information from the lattice structure.

We limit our discussion to the introductory information of these models. However, the interested readers can find more details of these models in a few authoritative references [2, 3, 5, 7, 13, 27, 32].

2.4. FCA for IR

Mathematical lattices are useful for conceptual IR. For IR tasks, a term-document matrix is treated as a formal context with objects as documents in the collection and attributes as the indexing terms. The lattice structure obtained from the term-document formal context contains the nodes composed of a subset of index terms, i.e., intent; and subset of documents, i.e., extent. The nodes are partially ordered and if the two nodes are comparable, then a link exists between them. The top node in the lattice structure contains all the documents defined by their common indexed terms.

The lattice structure helps the user to refine the query decreasing the cognitive overload on the user. A query formed by a set of attributes $\boldsymbol{A} \subseteq \boldsymbol{M}$ can be considered as a query concept. We can find documents similar to the query from the concepts whose intent is equal to the query or the concepts whose intent is subset of the query. In a different approach, the extent of the query concept contains the documents having query terms and the intent of the query concept contains all the terms that are possessed by the documents in the extent. From such a query concept one can use the nearest neighbor algorithm to find the related concepts.

Lattice representations were used in early IR literature for query refinement. A query can initially be submitted to the lattice based IR system to locate the precise answer concept. Once the answer is identified, additional results can be obtained by browsing the lattice. Subsequently thesaurus based concept lattices have been used for query refinement for more general or specific queries. Document ranking can be performed by using concept lattice structure for computing the conceptual distance between the query and each document. Concept lattices can be used to optimize the performance of hierarchical clustering based ranking. Systems like CREDO for mining the retrieval results returned by a web search engine are also developed using concept lattices (**http://credo.fub.it/**).

Lattice based retrieval models are available in literature since 1960's. Mooers model is the first application of lattices in IR applications. P o s h y v a n y k and M a r c u s [25] have combined FCA and LSI to address the problem of concept location in a source code, in which FCA is used to organize the results obtained from LSI. R a j a p a k s e and D e n h a m [29] have combined reinforcement learning with FCA for the retrieval of useful documents to the users' queries. They have also identified the disadvantages in the past approaches. P r i s s [28] has developed lattice based retrieval systems called FaIR that incorporate a graphical representation of a faceted thesaurus. Unlike data driven approach, a facet based IR system tries to control the lattice complexity by restricting the possible keyword combinations. D i a z-A g u d o and G o n z a l e z-C a l e r o [17] have presented FCA as a supporting technique for the case based retrieval. B e c k e r and E k l u n d [11] have studied the prospectus for document retrieval using FCA. They have studied FCA for query refinement, document ranking. They have also developed a web based retrieval system, known as Score. In another work, E k l u n d et al. [19] have used FCA for social tagging and IR applications of a digital library. N i z a r et al. [24] have proposed FCA based algorithm called BR-Explorer for IR. The FooCA system developed by Koester, applies the formal concepts to web retrieval. The

other FCA based conceptual knowledge processing tools include Docco, ToscanaJ, Tockit, Score. Recently F o r m i c a [21] has combined rough set theory and fuzzy FCA to perform semantic web search and discovery of information. P o s h y v a n y k and M a r c u s [25] and P o s h y v a n y k et al. [26] have combined LSI and FCA for concept location in a source code. A h m a d [1] has proposed image indexing and retrieval technique using FCA.

3. Experiments

In this section we conduct experiments on Medline standard IR dataset and a real world healthcare dataset.

3.1. Chronic bronchitis data

The healthcare dataset is a part of consumer healthcare informatics project of the Medical Research Council of South Africa [4]. The diseases which were studied in the project are TuBerculosis (TB), Chronic Bronchitis (CBr) and HyPertension (HP). However, in our analysis we consider only CBr data. For both Medline and CBr data we have conducted experiments using VSM, LSI and FCA. Chronic Bronchitis dataset contains data about 7 patients for various symptoms of CBr and experts' rules for determining the disease. Table 1 shows the list of various CBr symptoms. Table 2 lists experts' opinions in the form of rules for determining the disease using the symptoms listed in Table 1. Table 3 shows the formal context, also known as object-attribute binary incidence matrix of CBr data with the details of 7 patients. The last column of the matrix indicates treating doctors' conclusion on the presence or absence of CBr. Fig. 1 shows the concept lattice obtained by applying FCA on the CBr incidence matrix given in Table 3. The concept lattice shown in Fig. 1 is of height 6 and contains 10 concepts with 12 edges. We consider the patient symptoms listed in Table 1 as keywords and the objects are the documents. For illustration purpose we consider the experts' rules listed in Table 2 as queries over the documents.

Table 1. Chronic bronchitis symptoms

No	Symptom	Abbreviation
1	Persistent Cough	PC
2	Sputum Production	SP
3	Sputum produced is Muco Purulent	MC
4	Chest Tightness	CT
5	Shortness of Breath	SB
6	Wheezing Chest	WC
7	Smoking	SM

Table 2. Expert's rules for CBr

Sl. No	Expert Rules for Tuberculosis
1	PC SM SP CT SB → CBr
2	PC SM SP WC → CBr
3	PC SM SP MC → CBr

We can find that the root node of the Bronchitis lattice structure is the concept which has the set of all documents as its extent and the bottom node is the concept containing the set of all attributes in its intent. An edge between any two concepts in the lattice structure exists if they are comparable. Navigating the lattice structure downwards is known as specialization. Since the set of the keywords possessed by the documents increases, the documents having these keywords decreases. Similarly, when we navigate upwards in the lattice structure, the keywords in the intents decrease and the documents having these keywords increase. Hence, we call navigating the structure upwards specialization. This behaviour is called a principle of duality.

Table 3. Incidence matrix from original Bronchitis dataset

Object	PC	SP	MC	CS	CT	SB	WC	SM	CBr
Obj 1	X	X	X					X	
Obj 2	X							X	
Obj 3	X	X		X	X			X	
Obj 4	X	X	X		X	X	X	X	X
Obj 5	X	X		X	X		X	X	
Obj 6								X	
Obj 7								X	

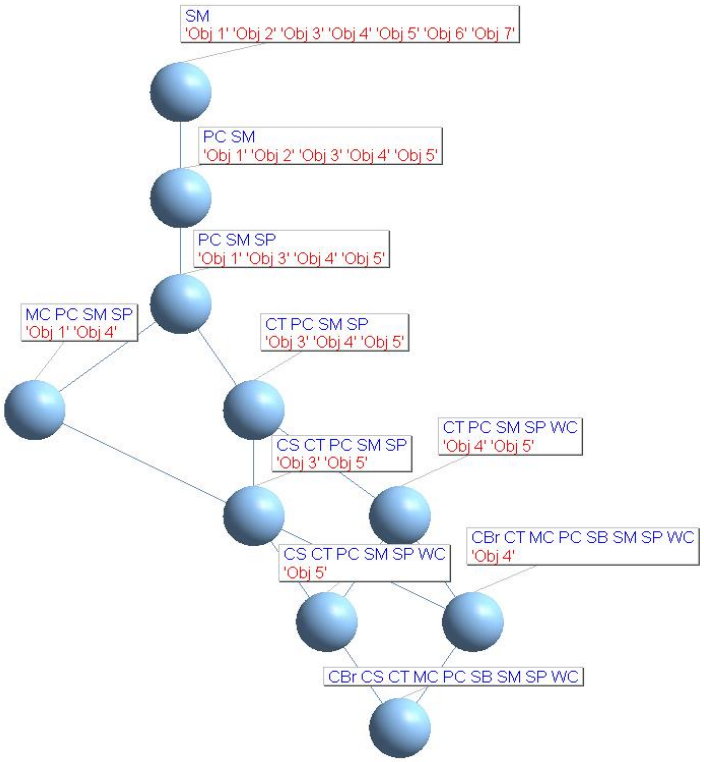

Fig 1. Lattice structure of CBR context shown in Table 3

We consider the query also as a concept document and the map with the lattice produced from the original context. This mapping will transform the lattice structure by adding or modifying the nodes with query concept. Fig. 2 shows the new CBr lattice structure obtained after mapping Query 1 (PC, SM, SP, CT, SB, CBr). As mentioned above, we have considered the experts' rules listed in Table 2 as queries. Then the document concepts are ranked based on the distance with regard to the number of edges from the query concept in the transformed lattice. The documents that were equally distant would receive the same rank. Table 4 lists the document concepts based on the rank. We can find that object 4 with keywords (symptoms: PC SP MC CT SB WC SM CBr) is close to the given query. Based on the distance from the query node, all other documents are also ranked. Similarly Figs. 3 and 4 display the lattice structures obtained after mapping Queries 2 and 3. Table 4 lists the ranks of the documents for these queries. For Query 3 (PC SM SP MC CBr), we can find from Table 3 that object 4 is the clear match. Though Object 1 is having the keywords (symptoms), since it has no CBr word, its rank is 2.

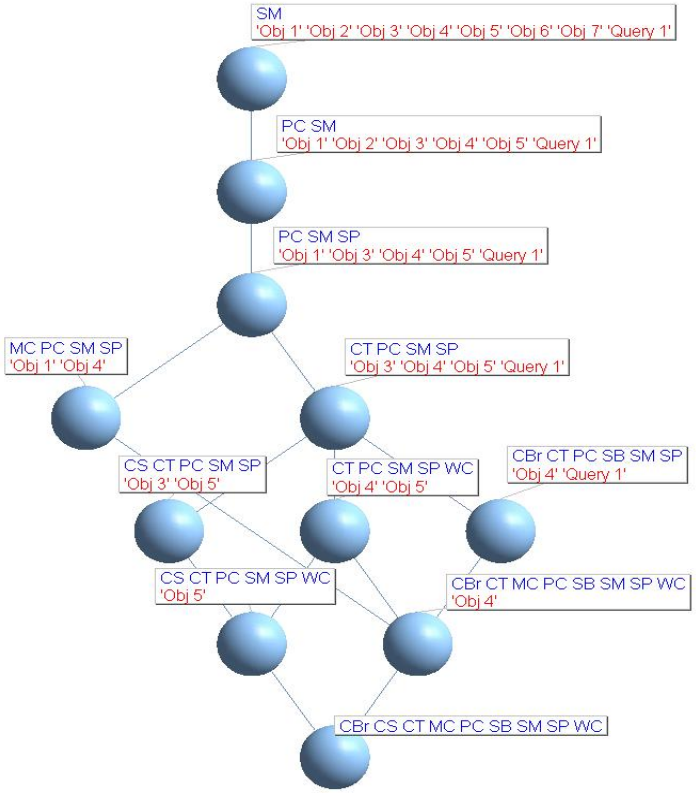

Fig 2. CBR Lattice with first expert rule as Query 1

Table 4. Ranking CBr documents using FCA

Rank	Q1	Q2	Q3
	Document	Document	Document
1	O4	O4	O4
2	O3, O5	O5	O1
3	–	O1, O3	O3
4	O1	–	O2, O5
5	O2	O2	–
6	O6, O7	O6, O7	O6, O7
7	–	–	

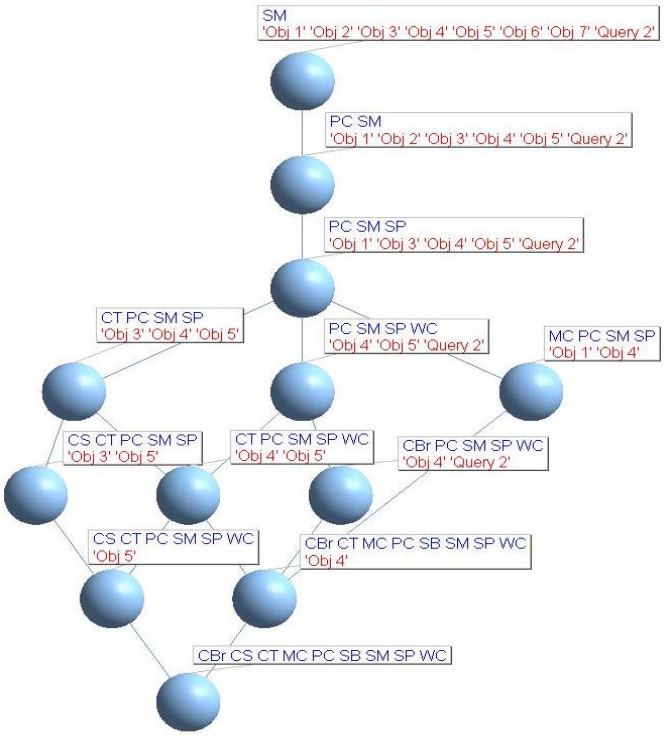

Fig 3. CBR Lattice with second expert rule as Query 2

Next we apply VSM, LSI on the CBr data matrix shown as context in Table 3. We consider the query vectors from Table 2. The experts' rules in Table 2 are transformed as query vectors as follows:

Expert rule 1: Query 1 =[1 1 0 0 1 1 0 1 1];

Expert rule 2: Query 2=[1 1 0 0 0 0 1 1 1];

Expert rule 3: Query 3=[1 1 1 0 0 0 0 1 1];

We have used cosine similarity measure for finding the similarity between the query vectors and the object documents in the CBr context. Table 5 lists the ranks of CBr object documents for the three queries using VSM along with the similarity score they have obtained. We can observe that the document ranking for query 1 is exactly similar to FCA. For Queries 2 and 3, the rankings are almost the same except at higher ranks.

Next we have applied LSI. For implementing LSI we have chosen the rank value (k = 3) and applied truncated SVD on the CBr formal context matrix of size 7×9. The queries are projected over the new low dimensional space and similarity is computed. Table 6 lists the document ranking along with the similarity values of each of the documents. We can notice from Table 6 that for Query 1, LSI has retrieved object 4 as the first document similarly to VSM and FCA. For Query 3, LSI retrieval results are almost similar to VSM results. However, for Query 2 the ranked results are different for VSM and FCA.

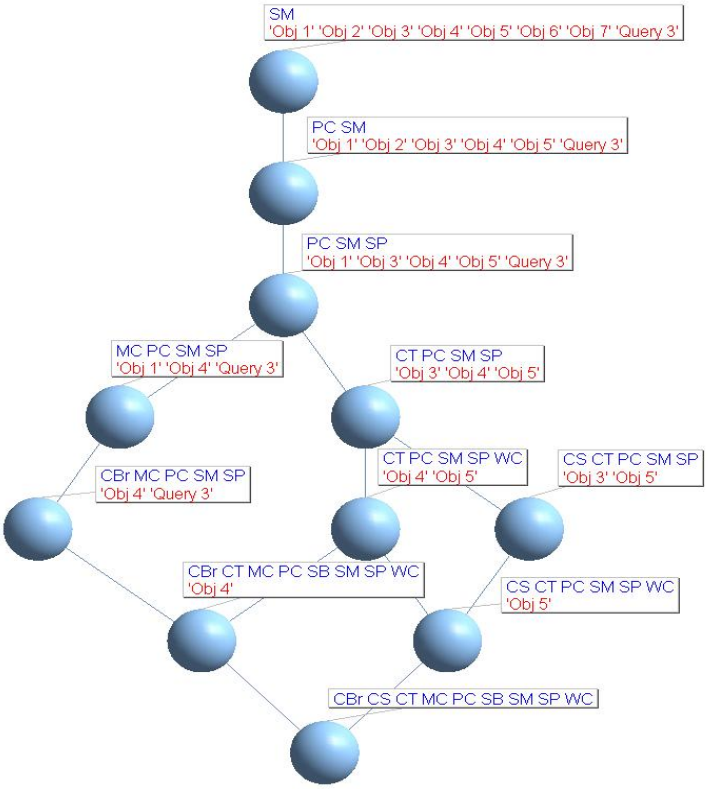

Fig. 4. CBR Lattice with the third expert rule as Query 3

Table 5. Ranking CBr documents using VSM

Rank	Q1		Q2		Q3	
	Document	Similarity	Document	Similarity	Document	Similarity
1	O4	2.123	O4	1.7678	O1	2.0
2	O3	1.7889	O5	1.6330	O4	1.7678
3	O5	1.6330	O1	1.5	O2	1.4142
4	O1	1.5	O2	1.4142	O3	1.3416
5	O2	1.4142	O3	1.3416	O5	1.2247
6	O6, O7	1.0	O6, O7	1.0	O6, O7	1.0
7	–	–	–	–	–	–

Table 6. Ranking CBr documents using LSI k=3

Rank	Q1		Q2		Q3	
	Document	Similarity	Document	Similarity	Document	Similarity
1	O4	1.7880	O1	1.6546	O1	1.9445
2	O1	1.7755	O4	1.6079	O4	1.7965
3	O5	1.7615	O5	1.5157	O2	1.4837
4	O3	1.7424	O3	1.5103	O5	1.2754
5	O2	1.4558	O2	1.3905	O3	1.2720
6	O6, O7	0.9477	O6, O7	0.9781	O6, O7	0.9884
7	–	–	–	–	–	–

3.2. Medline data

The Medline document collection contains totally 1033 Medical abstract documents indexed by 5735 terms. The dataset contains also 30 user queries and their relevance judgments. An important heuristic in indexing the document collection is the term weighting [31]. We have used the standard tf-idf term weighting method on Medline document collection. A detailed explanation on the term weighting methods is available in [6, 31 and references therein]. We have conducted retrieval experiments using FCA, VSM and LSI. Among the 30 queries available with Medline dataset, we have considered in this analysis the query numbers 1, 9, 20 and 29. Table 7 shows the relevant documents in Medline collection for these selected queries. An interesting result we can observe is that all the methods have ranked documents 6 &7 at rank 6.

Table 7. Relevant documents for selected queries

Query	Relevance documents
1	13 14 15 72 79 138 142 164 165 166 167 168 169 170 171 172 180 181 182 183 184 185 186 211 212 499 500 501 502 503 504 506 507 508 510 511 513
9	30 31 53 56 57 64 83 84 89 124 125 126 192 252 253 267 268 269 270 271 272 273 409 412 415 420 421 422
20	567 570 571 573 574 575 576 577 578 580 581 584 585 588 589 590 593 594 595 596 597 598 599 601 602 848 869 870 871 872 873 874 875 876 877 878 879 880 883
29	740 741 742 743 744 745 746 747 748 749 750 751 752 754 757 759 760 761 762 763 764 765 766 767 768 853 1004 1006 1007 1008 1009 1010 1012 1013 1015 1016 1017

First we apply FCA on Medline document collection. To perform this operation, we have discretized the entries in the term-document matrix based on a threshold value and hence the matrix becomes the crisp formal context. Then we have applied FCA to find the concepts. From all the concepts, we have identified the concepts having the given query in the concepts' extent. At the next step we have retrieved the documents in the matched concepts extent and the documents are ranked by counting the terms of the given query in each matched concept's intent. Table 8 lists the retrieval results of the selected queries using FCA.

Next we have applied VSM on the term-document matrix of Medline collection. We have used the cosine similarity metric to calculate the similarity

between the queries and the term weighted document collection. Table 9 shows the first 10 ranks of the retrieval results using VSM for the selected queries, along with the similarity score obtained for each document.

Table 10 shows the retrieved results of the selected queries by applying LSI on Medline document collection. While performing SVD on the term-document matrix, we have considered the reduced rank approximation value k = 100. Selecting the optimal value for this intrinsic dimensionality parameter is an interesting problem and depends on the characteristics of the document collection.

Table 8. Ranking Medline documents using FCA

Rank	Q1	Q9	Q20	Q29
	Document	Document	Document	Document
1	212	271	589	743
2	167	126	873	1008
3	14	84	590	750
4	72	420	589	1017
5	138	421	872	745
6	142	272	596	1004
7	165	252	584	740
8	168	415	880	742
9	180	267	573	1010
10	183	56	602	853

Table 9. Ranking Medline documents using VSM

Rank	Q1		Q9		Q20		Q29	
	Doc	Similarity	Doc	Similarity	Doc	Similarity	Doc	Similarity
1	72	0.7745	409	0.6334	878	0.9594	1012	2.5275
2	500	0.6467	415	0.6130	596	0.4147	1016	2.5016
3	15	0.4862	273	0.5250	577	0.7455	1008	2.4729
4	171	0.4339	268	0.4026	876	0.7273	1017	2.2212
5	513	0.4210	89	0.3935	873	0.7076	1015	2.1903
6	166	0.4022	30	0.3850	584	0.7028	853	2.1851
7	181	0.3918	422	0.3803	581	0.6974	1007	1.9206
8	168	0.3668	126	0.3472	874	0.6954	1009	1.5140
9	169	0.3477	267	0.3375	872	0.6325	740	1.5054
10	511	0.3417	56	0.3271	879	0.6286	1013	1.2576

Table 10. Ranking Medline documents using LSI, k = 100

Rank	Q1		Q9		Q20		Q29	
	Doc	Similarity	Doc	Similarity	Doc	Similarity	Doc	Similarity
1	212	38.65	126	26.2580	597	52.9194	1008	249.1960
2	499	22.87	420	19.4166	878	45.11	1017	160.7817
3	142	19.1480	422	18.6230	581	42.26	1009	94.4149
4	15	18.4684	421	17.3854	596	35.6636	853	94.2499
5	166	17.3576	267	15.8298	580	32.6934	1016	93.39
6	513	16.342	409	12.2383	590	29.0818	1013	84.7088
7	511	15.5151	271	11.7703	584	28.97	740	78.97
8	165	13.2578	64	11.5332	577	27.925	1015	69.2969
9	183	12.9967	412	11.1317	588	27.5048	1012	60.5185
10	182	11.6852	415	11.0308	570	26.99	1006	53.9709

We can observe from Tables 8, 9 and 10 and the relevance documents Table 7, that the top 10 documents retrieved using FCA, VSM and LSI are relevant to the concerned queries.

4. Discussion

We believe that the analysis provides good understanding for IR researchers on the effectiveness of VSM, LSI and FCA. The experiments on CBR data demonstrate how FCA can be applied in retrieval applications. Further experiments on Medline data demonstrate the results on a standard IR dataset. The results indicate that the retrieval performance of FCA and classical models is similar. An IR system is aimed at retrieving all, and only documents relevant to the user query or the information need. The notion of relevancy is having important significance in IR since it helps the user in narrowing the search space. Relevancy of the document to a query depends on the context of the users' need. The classical keyword based retrieval systems do not take into account the context of the user queries and the content of the documents indexed. LSI captures the semantic information hidden in the term-document matrices through dimensionality reduction using SVD. Hence, LSI derives the semantic similarity between the user information need and the documents in the collection. An important issue is the scalability. The vector based models prove to be applicable on large size document collections, such as TREC. Though computational requirements of SVD are high in LSI, alternative strategies have evolved in literature to mitigate the problem [9, 10 and references therein].

In IR, the mathematical lattices are more applicable for representing conceptual hierarchies. A concept lattice can be obtained from term-document matrix by transforming it to a formal context. The query terms are mapped onto the lattice structure and if a node shares the same keywords with the query terms, then the document is relevant to the query and hence it is retrieved. The principle of duality that exists between the extent and intent of the formal concept forms a Galois connection between the two. This principle makes FCA more suitable for IR applications, since a smaller set of keywords returns a larger document set and a larger set of keywords returns a smaller document set. In contrast to the other hierarchical approaches like trees, similar documents will be always close in the lattice structure. Each node represents a cluster of documents sharing common attributes.

Additional keywords in the query move the query concept down the current node which maps mathematically to the definition of a subconcept relation. Documents that are sharing similar attributes will be located closely in the lattice. Moving upwards in the lattice structure from the query concept, results in losing at least one match. The documents at the query node are better matches than the documents at the sub concepts.

On Medline collection while applying FCA, we have retrieved extents of the concepts whose intents matches with the intents in the user given query concept. Matching by means that we retrieve the extent of the concepts whose intent is a subset of intent of the given query. The largest subset receives the first rank and so

on. In order to perform this operation, the retrieval algorithm should scan all the concepts of the given context. The worst case time complexity of this process is $O(2^n)$ since the maximum number of concepts that can be generated from n number of attributes is 2^n. The number of steps required for analysis is linear with the number of concepts generated. However, while applying FCA on CBr data, we have used the attribute labels for decision making. This approach is to demonstrate one of the advantages of using concept lattices.

Despite the advantage of efficiently representing knowledge, lattice construction is not an easy task. The number K of nodes in a concept lattice has linear complexity with the number m of documents in the collection $H = O(m.2^n)$, where O denotes the computational complexity and n denotes the maximum number of terms [3, 8, 15]. The concept generation in FCA is a recursive procedure and several algorithms are available in literature [13]. Few researchers are aiming to solve this scalability issue in lattice based retrieval by proposing alternative hybrid methods. The term-document space can also be treated as a many valued context in FCA scenario. Weighted frequencies of the words are the attribute values. Through conceptual scaling, the multi-valued context can be transformed into a single context. However, the choice of scaling is based on the need, requiring interpretation and is generally performed by the domain expert.

For developing conceptual information systems and text mining, FCA can be combined with other popular techniques like LSI, CI, etc. [15, 10, 26]. In retrieval applications, either the search space or the search results can be clustered using FCA. While applying FCA to cluster the search results, P o s h y a n k et al. [26] have tackled the problem of scalability in FCA by applying it only on the subset of relevant search results. On the other hand, the matrix decompositions of LSI model can be used for handling the complexity of FCA tasks [4, 7, 8, 30]. An interesting recent investigation on extending fuzzy FCA for handling the weighted documents and queries can be found in [34]. Future directions were discussed in [26] while combining FCA with other IR approaches.

5. Conclusions

This paper is aimed at providing an insight into IR using the classical vector model, its variant LSI and mathematical lattice based FCA. Experiments are conducted on the standard Medline and real world healthcare datasets. Our analysis shows that the retrieval performance of these models is similar except minor changes in their ranking of the relevant documents. Future research will mainly concentrate on integrating the hierarchical knowledge representation models, like FCA with IR models in further optimizing the retrieval performance.

Acknowledgements: Authors Ch. Aswani Kumar & J. Annapurna sincerely acknowledge the financial support of the National Board of Higher Mathematics, Dept. of Atomic Energy, Govt. of India under Grant number 2/48(11)/2010-R&D II/10806.

References

1. Ahmad, I. S. Text-Based Image Indexing and Retrieval using Formal Concept Analysis. – KSII Transactions on Internet and Information Systems, Vol. **2**, 2008, No 3, 150-170.
2. Kumar, A. C., S. Srinivas. Latent Semantic Indexing Using Eigen Value Analysis for Efficient Information Retrieval. – International Journal of Applied Mathematics and Computer Science, Vol. **16**, 2006, No 4, 551-558.
3. Kumar, A. C., S. Srinivas. Concept Lattice Reduction Using Fuzzy *k* Means Clustering. – Expert Systems with Applications, Vol. **37**, 2010, No 3, 2696-2704.
4. Kumar, A. C., S. Srinivas. Mining Associations in Health Care Data Using Formal Concept Analysis and Singular Value Decomposition. – Journal of Biological Systems, Vol. **18**, 2010, No 4, 787-807.
5. Kumar, A. C., S. Srinivas. On the Performance of Latent Semantic Indexing Based Information Retrieval. – Journal of Computing and Information Technology, Vol. **17**, 2009, No 3, 259-264.
6. Kumar, A. C., S. Srinivas. A Note on the Effect of Term Weighting on Selecting Intrinsic Dimensionality of Data. – Cybernetics and Information Technologies, Vol. **9**, 2009, No 1, 5-12.
7. Kumar, A. C. Fuzzy Clustering Based Formal Concept Analysis for Association Rules Mining. – Applied Artificial Intelligence, 2012.
8. Kumar, A. C. Knowledge Discovery in Data Using Formal Concept Analysis and Random Projections. – International Journal of Applied Mathematics and Computer Science, Vol. **21**, 2011, No 4, 745-756.
9. Kumar, A. C. Analysis of Unsupervised Dimensionality Reduction Techniques. – Computer Science and Information Systems, Vol. **6**, 2009, No 2, 217-227.
10. Kumar, A. C., S. Srinivas. A Note on Weighted Fuzzy *k*-Means Clustering for Concept Decomposition. – Cybernetics and Systems, Vol. **41**, 2010, No 6, 455-467.
11. Becker, P., P. Eklund. Prospectus for Document Retrieval Using Formal Concept Analysis. – In: Proceedings of 6th Australasian Document Computing Symposium, 2001.
12. Berry, M. W. et al. Matrices, Vector Spaces, and Information Retrieval. – SIAM Review, Vol. **41**, 1999, No 2, 335-362.
13. Carpineto, C., G. Romano. Concept Data Analysis: Theory and Applications. John Wiley and Sons, 2004.
14. Carpineto, C., G. Romano. Using Concept Lattices for Text Retrieval and Mining. Formal Concept Analysis: Foundations and applications, LNCS 3626, 2005, 161-179.
15. Codocedo, V., C. Taramasco, H. Astudillo. Cheating to Achieve Formal Concept Analysis over a Large Formal Context. – In: Proceedings of the Concept Lattices and Their Applications, 2011, 349-362.
16. Deerwester, S., S. Dumais, G. Furnas, T. Landauer, R. Harshman. Indexing by Latent Semantic Analysis. – Journal of American Society for Information Science, Vol. **41**, 1990, No 6, 391-407.
17. Diaz-Agudo, B., P. A. Gonzalez-Calero. Formal Concept Analysis as a Support Technique for CBR. – Knowledge Based Systems, Vol. **14**, 2004, 163-171.
18. Dominich, S. The Modern Algebra of Information Retrieval. Springer, 2008.
19. Eklund, P., P. Goodall, T. Wray. Information Retrieval and Social Tagging for Digital Libraries using Formal Concept Analysis. – In: Proceedings of International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future, 2010, 1-6.
20. Evangelopoulos, N., Z. Zhang, V. R. Prybutok. Latent Semantic Analysis: Five Methodological Recommendations. – European Journal of Information Systems, Vol. **21**, 2012, 70-86.
21. Formica, A. Semantic Web Search Based on Rough Sets and Formal Concept Analysis. Knowledge Based Systems, 2011.
22. Ganter, B., R. Wille. Formal Concept Analysis: Mathematical Foundations. Berlin, Springer, 1999.

23. H o s s a i n, M. M., V. P r y b u t o k, N. E v a n g e l o p o u l o s. Casual Latent Semantic Analysis: An Illustration. – International Business Research, Vol. **4**, 2011, No 2, 38-50.
24. N i z a r, M., M. D. D e v i g n e s, A. N a p o l i, M. S m a i l-T a b b o n e. BR-Explorer: An FCA Based Algorithm for Information Retrieval. – In: Proceedings of 4th International Conference on Concept Lattices and Their Applications, 2006, 285-290.
25. P o s h y v a n y k, D., A. M a r c u s. Combining Formal Concept Analysis with Information Retrieval for Concept Location in Source Code. – In: Proceedings of the 15th IEEE International Conference on Program Comprehension, 2007, 37-48.
26. P o s h y v a n y k, D., M. G e t h e r s, A. M a r c u s. Concept Location using Formal Concept Analysis and Information Retrieval. ACM Transactions on Software Engineering and Methodology, 2012 (to appear).
27. P r i s s, U. Formal Concept Analysis in Information Science. – Annual Review of Information Science and Technology, Vol. **40**, 2007, 521-543.
28. P r i s s, U. Lattice Based Information Retrieval. – Knowledge Organization, Vol. **27**, 2000, No 3, 132-142.
29. R a j a p a k s e, R. K., M. D e n h a m. Text Retrieval with More Realistic Concept Matching and Reinforcement Learning. – Information Processing and Management, Vol. **42**, 2006, 1260-1275.
30. S n a s e l, V., P. G a j d o s, M. D. A. H u s s a m, M. P o l o v i n c a k. Using Matrix Decompositions in Formal Concept Analysis. – In: Proceedings of 10th International Conference on Information System Implementation & Modeling, 2007, 121-128.
31. S r i n i v a s, S., A. C. K u m a r. Optimizing Heuristics in Latent Semantic Indexing for Effective Information Retrieval. – Journal of Information and Knowledge Management, Vol. **5**, 2006, No 2, 97-105.
32. S t u m m e, G. Formal Concept Analysis. Handbook of Ontologies. Springer, 2009, 177-199.
33. W a n g, Q., J. H u, H. L i, N. C r a s w e l l. Regularized Latent Semantic Indexing. – In: Proceedings of the 34th ACM SIGIR Conference on Research and Development in Information Retrieval, 2011, 685-694.
34. Y a s s i n e, D. Extended Galois Derivation Operators for Information Retrieval Based on Fuzzy Formal Concept Lattice. – In: Proceedings of Scalable Uncertainty Management, LNAI 6929, 2011, 346-358.
35. Y a t e s, R. B., B. R. N e t o. Modern Information Retrieval. New Delhi, Pearson Education, 2004.