ISSN:2320-0790

An International Journal of Advanced Computer Technology

Automatic Indexing Framework for Context Aware Personal Document Management System

Y.D. Jayaweera[1], Md Gapar Md Johar[2], S.N. Perera[3]
[1]*Management and Science University, Malaysia*
[2]*Management and Science University, Malaysia*
[3]*University of Colombo, Sri Lanka*

Abstract: Today managing files in a personal computer has the same magnitude as managing the World Wide Web due to the dynamic nature of the file system [1]. Even searching for files over a file system is time consuming because finding a file on hard disk is a long-running task. Every file on the disk has to be read with dangling pointers to files which no longer exist because they have been changed, moved or deleted. This makes the user frustrated. Personal document management is the activity of managing a collection of digital documents by the owner of the documents.

This consists of creation, organization, finding and maintenance of documents. Information, especially digital information, is no longer a scarce resource; information exists in abundance and human time and attention have now become the scarce resource [2]. Information overload is now a recognized problem as people struggle to manage the increasing quantities of information they need to deal with on a daily basis [3]. Therefore, an on demand software agent comes necessary to manage the file system and retrieve information that the User needs.

This research proposes a framework that manages semi-structured file collection based on Formal Concept Analysis (FCA). Formal Concept Analysis has been applied in document retrieval in different contexts. Solutions like Conceptual Email Manager [4] and DOCCO [5] do not support dynamic insertions of documents (Extents) to their concept lattice. These tools require algorithms to re-run in order to rebuild the index which is costly. In this work concept lattice is generated incrementally on a document collection and stored in a database making the hierarchical structure compact to facilitate parallel insert and search. The proposed framework utilizes database queries and functions complemented with inverted index which facilitates fast document retrieval and reduce the downtime of costly updates on the master index.

Keywords: Information Retrieval, Personal Document Management, Formal Concept Analysis (FCA), Incremental Formal Concept Analysis, Context Aware Document Retrieval, Automatic Indexing, Document Clustering.

1. INTRODUCTION

In the early days of human development, stone and clay tablets, animal bones and shells and various metals played a role in the preservation of documents. But then paper was invented in China during the Han dynasty. And for the next 2,000 years, physical paper documents were the dominant medium for all written and recorded information [6]. Today electronic documents consisting of bits and bytes replace paper. One such example is how people tend to manage their documents. Almost all the personal files are stored in personal computers in digital format. Not only personal documents but also wallets and keys are stored in digital format making it accessible at their fingertips.

There are many benefits of having electronic documents which include reducing storage, easy indexing, flexible searching, fast document distribution and managed security. Because of rapid growth in technology and innovation people tend to put their faith in the digital world more than ever. Devices like smartphones, PDAs, personal computers and e-book readers also leverage the use of electronic documents.

But due to the flood of information in personal computers the problem of information overload must be addressed with better decision support and data processing tools [7]. The challenge of an effective document management system in this ever growing digital environment is to

manage and use electronic documents effectively. The Users spend a great deal of their time using software tools to locate and manage files stored on their personal computers. The sheer scale of files makes the finding right content extremely difficult [3]. It leaves Users dangling in the digital environment. Information overload is now a recognized problem as people struggle to manage the increasing quantities of information they need to deal with on a daily basis [2].

Due to an excess of files available on a personal computer Users are required to do a proper filing and structuring of digital files. But more often than not, Users forget to do a proper structuring or filing allowing files to pile in temporary folders or even on their desktop making it hard to locate and manage. On the other hand, with a proper filing and structuring it yields significant reduction of time in locating and managing files. It is important that the software tools with a proper design should effectively support document management activities.

The hierarchical file system used today to manage documents has been there for decades without any conceptual change [8]. Even though there are some working prototypes developed, these have not been successful due to lack of contextual understanding of User document management behavior. In this study an automatic file indexing framework using Formal Concept Analysis was proposed to bridge the gap between Users' perceived information need and the generic search result which does not count Users' context. Further, the work strengthens the application of FCA in document retrieval by enabling dynamic document insertions on the run. The proposed framework is implemented as a knowledge layer on top of the existing file system without changing its physical structure. It leaves the User the ownership of his file structure [8]. It builds concepts incrementally on semi-structured text documents as to show such a solution is viable with improved precision.

The rest of the paper is organized as follows. Section 2 gives an overview of current status of personal document management systems, Information Retrieval and examines the use of FCA for information retrieval and discusses the advantages of FCA for context aware document browsing. Section 3 presents the discussion of the proposed framework. Finally Section 4 concludes the work with topics for future direction.

2. LITERATURE REVIEW

Personal Document Management has several definitions. Among many definitions, for this study Personal Document Management is defined as "The organization and maintenance of personal document collections in which information items, such as paper documents, electronic documents, email messages, web references, handwritten notes, etc., are stored for later use and repeated re-use" [9].

According to the definition, Personal Document Management focuses on activities related to organization and maintenance of information collections. All modern personal computer operating systems have a hierarchical file system consisting of virtual folders which contains both User and System files [3]. The file system accommodates several types of files which consist of executable files, xml files, text documents, spreadsheet files, presentation files, HTML files, PDF files, email documents, audio files, video files, images, memos, contacts etc. Each and every file type has its own format making it harder to extract metadata in a uniform manner. A file type is a group of reusable settings that describe the shared behaviors for a specific file type and requires a handler to process it. A file handler knows particular features and requirements of each of these different file types. This behavior makes it difficult to process different file types in a uniform manner.

Personal Document Management (PDM) has three key areas: Filing, Organizing and Finding [9]. Filing refers to creating an appropriate folder structure and storing of files. With a proper filing strategy users can locate the file without consuming much time. Organizing means creating meaningful and deep document structures similar to a mind map. Related folders are grouped together to create more meaningful structures and to hide complexities. Folders are arranged in such a manner that one can extract the context through the folder structure. Finding implies locating files required by the User. The key areas are highly interconnected. Previous research shows that there is a tradeoff between filing and finding. The higher filing effort yields lesser finding effort [9]. But due to sheer volume of files and time constraints Users sometimes may not adhere to a proper filing strategy and this makes locating a file very hard at a later stage.

There are a number of tools developed to assist Users in Personal Document Management activities which yield to manage and find files on their personal desktops. The tools can be categorized as PDM Suites, Mind mapping tools, tools for Note taking, Desktop search tools and Semantic PDM tools. PDM Suites such as MS Outlook, Lotus Notes and the like focus mainly on storage and they lack context [10]. In such tools files are stored somewhere but they do not come out when the User needs them. Mind mapping tools like MindManager and Freemind allow Users to connect structures and build a cognitive model but such tools lack searching and navigation [10]. Note-taking tools like OneNote help pillars more than filers. Such tools can capture information quickly but have weak support for structured data, making it difficult to map User's context [10]. Desktop search tools like Google Desktop Search is a local search engine that facilitates file search and background indexing.

Operating systems like Windows Vista and Windows 8 and 7 [11] facilitate search as an inbuilt feature in the OS. The search result can be saved for later reference. The stored

file saves the query which was used to search the files in XML format. When the user runs the stored XML file, the operating system re-runs the query on the Windows search subsystem. Their primary objective is to locate the file fast but no context, no hierarchy, no organization and no ranking. There are systems that remember files a user has visited and allows a user a quick access to the files recently visited. A related approach under this is the "Stuff I've Seen" system [12], which simply remembers all entities including files, Web pages, emails, contacts, etc. that a User uses on a computer. But like in the previous methods this lacks semantics which has no context. On the other hand revolutionary semantic PDM tools like Nepomuk-KDE connects words with a meaning and associate documents [7]. Semantic PDM tools leverage PDM activities by superior managing and locating files due to semantic rich data associated with it. seMouse is a another semantic desktop tool [13]. It introduces the notion of a knowledge folder as a coarse set of documents bound together by a common ontology. These semantic PDM tools require manual tagging in order to build and tune the knowledgebase which consumes additional amount of time and effort for the Users.

Considering the features and limitations of the existing tools for Personal Document Management it is observed that to bridge the gap of connecting words with meaning and to build association with documents automatically a knowledge layer is needed. To build a knowledge layer on top of the existing file structure Formal Concept Analysis is proposed due to its inherent hierarchical structure similar to User's cognition. This study focuses on building a framework to facilitate filing and finding of semi-structured files in a personal document collection. The proposed framework facilitates logical structuring of semi-structured file collection incrementally into a concept lattice to assist filing and facilitate finding of documents through the search API that uses an index built automatically.

2.1. INFORMATION RETRIEVAL

Information retrieval focuses on finding relevant information from a collection of information resources efficiently. Inverted file indexing has been widely used in information retrieval [14]. An inverted file is used for indexing a document collection to leverage the searching process without going through the entire document collection. The structure of an inverted file consists of two components: the vocabulary and the posting list. The vocabulary is composed of all distinct terms in the document collection. For each term, a list of all the documents containing this term is stored. The set of all these lists is called the posting list. However, the limitation of this approach is that the structure of a document is not counted in this model.

2.2. CONTEXT AWARE INFORMATION RETRIEVAL

Context can be defined as any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [15]. A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevance depends on the user's task [15].

Taking context into account is crucial when solving many information retrieval tasks in order to produce intuitive results and ultimately enable cognitive information retrieval [16]. Context aware information retrieval enhances finding of information through associating documents with their meaning. While adopting existing approaches of cognitive information retrieval for information retrieval reach high precision and recall rates, they fail to adapt their results to user preferences and external influences.

2.3. FORMAL CONCEPT ANALYSIS FOR INFORMATION RETRIEVAL

Formal Concept Analysis (FCA) is based on a mathematical framework for analyzing and structuring a domain of interest [17]. FCA can serve as a guideline for context building because it allows the identification of concepts by factoring out their commonalities while preserving concept specialization relationships. Due to this relationship between concepts in a concept lattice, it provides scaling of context at different levels of granularity.

FCA transforms a formal context into a concept lattice. A formal context is a representation of the relation between objects and their attributes. Formally, a context is a triple k = (G, M, I), where G is a set of objects, M is a set of attributes, and I is a binary relation $I \subseteq G\ X\ M$. Given a set of objects $A \subseteq G$, the shared image of A in M is defined as:

$$A\uparrow = \{m \in M | (g,m) \in I\ \forall\ g\ \in A \quad (1)$$

Similarly, for a set of attributes B ⊆ M, its shared image in G is:

$$B\downarrow = \{g \in G | (g,m) \in I\ \forall\ m\ \in B \quad (2)$$

$A \subseteq G, B \subseteq M$ and A= B↓, B= A↑. A is called the extent of the concept and B is called the intent of the concept [17]. In other words, equation 1 defines the collection of all attributes shared by all objects from A, and Equation 2 defines the collection of all objects sharing all the attributes from B. A partial ordering can be defined over the concepts of a context. Specifically,

$$(A1,B1) \leq (A2,B2) \leftrightarrow A1 \subseteq A2$$

Equivalently

$$(A1,B1) \leq (A2,B2) \leftrightarrow B1 \supseteq B2$$

The set of all formal concepts of a given context with the sub concept-super concept relation ($\leq$) is always a complete lattice, called the concept lattice.

The significant advantage of applying FCA in information retrieval is that the mathematical formulas of FCA can construct the conceptual structure which has generalization and specialization relationships among the concept nodes automatically [18][19]. In this approach, a document is annotated with a set of keywords which is then used for lattice generation. Concept on each node defines the context which associates bag of words (Intents) with the documents (Extents). This lattice structure allows User to reach a group of documents via one path, but then rather than going back up the same hierarchy and guessing another starting point (random seeker model), one can go to other parents of the present node improving the problem of category mismatch.

To develop accurate Lattice it is required to have good representation of keywords (bag of words) of the document collection and another problem of FCA is the memory consumption. Since most of the FCA algorithms are memory based the scalability of FCA to support large collection is limited. In this study the topics are extracted from semi-structured memoranda and used for concepts generation.

2.4. INCREMENTAL CONSTRUCTION OF THE CONCEPT LATTICE

Once the concept lattice is built it is required to maintain it with the changes. Specially, in the context of a personal document management system there will be lots of new documents added into the collection. Considering its dynamic nature, it is necessary to generate only the completed pairs of the lattice without having to regenerate it from scratch.

Incremental algorithms outperform most of the batch algorithms [20][21][22][23]. AddIntent algorithm is an approach to update formal concepts and concept lattice incrementally [5]. It was done by adding a new document with a set of keywords or by refining the keywords for an existing document.

AddIntent uses a similar methodology based on Propositions 1 and 2 below to avoid the consideration of old concepts and non-canonical generators in the lattice in a bottom-up search for canonical generators and modified concepts [5].

Proposition 1. If (B', B) is a canonical generator of a new concept (F', F), while (D', D) is a non-canonical generator of (F', F) – in this case, $B \subset D$ – then any concept (H', H) such that $H \subset D$ and $H \not\subset B$ is neither modified nor is a canonical generator of any new concept.

Proposition 2. If (D', D) is an old concept and $D \cap g' = B$ – in this case, $(B', B) \in L_i$ is modified – then any concept (H', H) such that $H \subset D$ and $H \not\subset B$ is neither modified nor is a canonical generator of any new concept.

This algorithm relies on a two-phased iterative approach to first discover canonical generators and in the second phase to update the lattice. AddIntent combines these two phases. In the proposed framework a similar kind of approach is adopted due to its novelty. The proposed framework uses a relational database management system to store, build and retrieve concepts to facilitate parallel insert and fast search. The entire concept lattice is stored and maintained by a relational database management system.

3. PROPOSED FRAMEWORK

The proposed framework in figure 1 functions in two ways. Firstly, the framework builds ordered concepts using the terms used in the Subject field of the semi-structured document (figure 2) as part of filing process. The complete lattice is stored in a relational database management system. AddIntent [5] algorithm is used to incrementally insert concepts to a relational database management system (RDBMS). As the corpus, a collection of memoranda are used for indexing. In the study a collection of semi-structured documents are used to highlight the importance of high precision required in a context aware information retrieval system and to show the effectiveness of using semantics in file structuring. Secondly the search interface, when user requests files through the Search API the framework returns recommended documents through searching the ordered concept's Intent and retrieving its Extent. The proposed system extracts concepts from structured documents, so that documents which are similar will be clustered together and then map search queries to concepts to recommend the documents.

The key design goal of the proposed framework is to handle document insertions without rebuilding the concept lattice, which is considered costly. The framework also utilizes the application of RDBMS in storing the complete concept lattice to facilitate parallel insertion and fast document search.

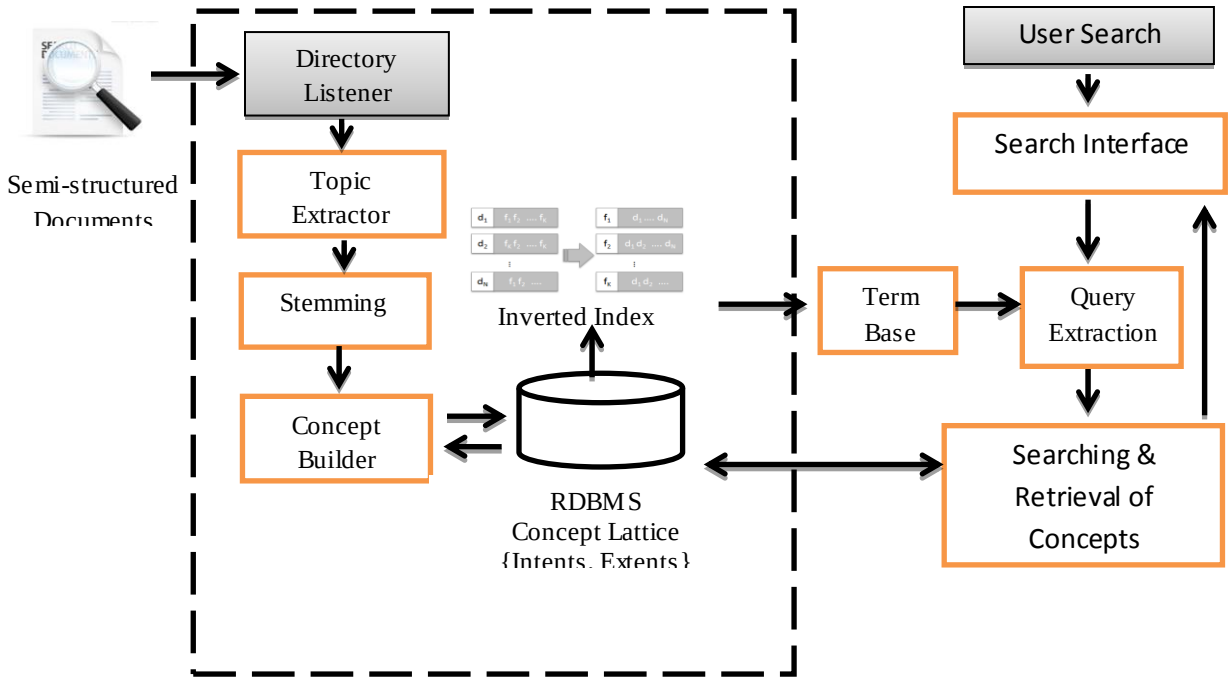

Figure 1: Framework

The following section describes the functionality of each component.

3.1. DIRECTORY LISTENER

The Directory Listener component provides the main interface that interacts with the physical file system. In the proposed framework it is required to register a specific root of the file system before receiving notifications of the file system changes. This gives flexibility to enable or disable the automatic indexing of the target file system. Once a directory is registered with the Directory Listener, it continuously receives document insertions in the target file system. The component intercepts the notifications and queues them for further processing.

3.2. TOPIC EXTRACTOR

The Topic Extractor component plays an important role in identifying document type and passing it into an appropriate handler. The document type handler reads the content of the document and extracts a bag of words which are best suited for the document. For the prototype as the corpus, a collection of memoranda are used for indexing (figure 2). The handler extracts Subject field to generate the bag of words.

INTEROFFICE MEMORANDUM

TO :
FROM :
SUBJECT :
DATE :

Figure 2: Semi-structured Document

The extracted Subject fields are tokenized and stop words are removed before term normalization begins. At the end of this process stemming algorithm runs to perform a process of linguistic normalization, in which the variant forms of a word are reduced to a common form. The normalization helps to reduce the number of terms used to represent the bag of words.

3.3. CONCEPT BUILDER

The Concept Builder component implements a modified version of AddIntent algorithm where the algorithm uses a database to store and retrieve concept lattice in contrast to the original AddIntent where it uses memory to store concept lattice. The component receives notification of document insertions where it runs Incremental FCA algorithm to maintain concept lattice and total ordering. The component uses database schema given in figure 3 to store and maintain Intents and Extents.

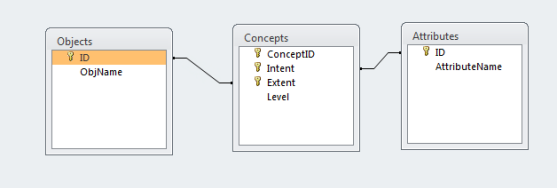

Figure 3: Database Schema

3.4. SEARCH INTERFACE

The main use of the Search Interface component is to intercept file retrieval requests from Users and facilitate index lookup to locate the stored files in the file system. The component performs a lookup using a term base to structure representation of the query based on the document collection. The term base can be automated by using a dictionary such as WordNet to identify synonyms. The component generates document frequency as the User navigates through the concepts. The support of a concept in the lattice is used to filter and rank documents. The Concept Lattice can be used to create a ranking on documents and to find documents that do not precisely match a query. Whenever two documents have similar attributes they will be located in concepts close together in the lattice. Once these distances are calculated the documents can be presented ordered by their rank. Unlike in classical approaches used in query expansions and document ranking, in the proposed framework the whole process is visible to the user and there are no heuristics involved.

4. SUMMARY AND FUTURE WORK

In this paper an Automatic Indexing Framework for Context Aware Personal Document Management System, which generates an incremental concept lattice automatically, was presented. As the text corpus, a collection of memoranda were used to extract semi-structured data. The extracted subjects given in the memoranda were used as a bag of words and to generate the concepts. The concept lattice was then stored in a relational database management system. The framework receives on demand document search requests from Users and is able to return the relevant files through the Search Interface. The Search Interface uses a Term base to identify synonyms. The generated output is validated through minimum support of the concepts. The support of a concept in the lattice is also used for document filtering and ranking.

The Automatic Indexing Framework for Context Aware Personal Document Management System is an effective solution for dynamic collections. It incorporates new concepts without rebuilding the entire concept lattice leaving the index up to date. A working prototype is to follow to prove the application of the proposed framework is effective with context aware Personal Document Management System. Another direction of interest is to accommodate automatic topic extraction into the framework.

Storing full concept lattice in the database facilitates parallel insertions and concurrent access to concepts. Further, an inverted index is maintained to fetch relevant tuples quickly from the database. The clustering technique utilized in the proposed system utilizes the fact that a given document can be attached to many topics avoiding one leveled index where a document belongs to a single cluster. The hierarchical ordering of concepts facilitates a document to be attached in multiple clusters. It makes automatic structuring and indexing one step closer to human reasoning.

REFERENCES

[1] Jayaweera, Y. D. 2012. Automatic File Indexing Framework: An Effective Approach to Resolve Dangling File Pointers. International Journal of Computer Applications 49(15):6-11, July 2012. Published by Foundation of Computer Science, New York, USA.

[2] Simon, H. A. 1997. The future of information systems. Annals of Operations Research, 71, 3-14.

[3] Edmunds, A., & Morris, A. 2000. The problem of information overload in business organisations: a review of the literature. Int. J. of Information Management, 17-28.

[4] Cole, R., and Stumme, G. 2000. CEM: A Conceptual Email Manager. In B. Ganter and G. Mineau (editors), Conceptual Structures: Logical, Linguistic, and Computational Issues, number 1867 in LNAI, pages 438–452. Springer Verlag, Berlin–Heidelberg–New York.

[5] Dean van der Merwe, Sergei, A. O. & Derrick, G. K. 2004. AddIntent: A New Incremental Algorithm for Constructing Concept Lattices. ICFCA 2004: 372-385.

[6] Microsoft, 2007. Introduction to Document Management. Retrieved January 21, 2014, from http://office.microsoft.com/en-001/sharepoint-server-help/introduction-to-document-management-HA010241399.aspx.

[7] KDE e.V. 2014. Nepomuk. Retrieved January 26, 2014, from http://userbase.kde.org/Nepomuk

[8] Elsweiler, D., Ruthven,, I., & Jones, C. 2005. Dealing with fragmented recollection of context in information management, Context-Based Information Retrieval (CIR-05) Workshop in CONTEXT-05.

[9] William, J. 2007. Keeping Found Things Found: The Study and Practice of Personal Information Management Retrieved January 26, 2014, from

http://searchdatamanagement.techtarget.com/feature/Personal-information-management-History-and-details.

[10] Kirchner & Bharti 2009. Mind map your way to an idea. Writer, Vol. 122(3), 28.

[11] Windows Search. Retrieved January 26, 2014, from http://windows.microsoft.com/en-US/windows7/products/features/windows-search.

[12] Dumais, S., Cutrell, E., Cadiz J., Jancke G., Sarin. R., and Robbins, D. C. 2003. Stuff I've seen: a system for personal information retrieval and re-use. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval (SIGIR '03). ACM, New York, NY, USA, 72-79.

[13] Iturrioz, J., Diaz, O., and Anzuola, S. F. 2008.Toward the Semantic Desktop: The seMouse Approach. IEEE Intelligent Systems, vol. 23, no. 1, pp. 24-31.

[14] Baeza-Yates, R., & Ribeiro-Neto, B. 1999. Modern information retrieval. New York: ACM Press.

[15] Dey, A. K. 2001. Understanding and Using Context. Personal Ubiquitous Computing, 5 (1):4–7.

[16] Spink, A., and Cole, C. 2005. New Directions in Cognitive Information Retrieval. Springer.

[17] Ganter, B., and Wille, R. 1989. Conceptual Scaling, In: F. Roberts (ed.): Application of Combinatorics and Graph Theory to the Biological and Social Sciences, Springer, 139-167.

[18] Becker, P., & Cole, R. 2003. Querying and analysing document collections with Formal Concept Analysis.

[19] Ganter, B., and Wille, R.1999. Formal Concept Analysis: mathematical foundations. Springer, Heidelberg.

[20] Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. 1983.Statistical semantics: analysis of the potential performance of key-word information systems, Bell System Technical Journal, 62, 1753-1806.

[21] Godin, R., Missaouri, R., and Alaoui, H. 1991. Learning algorithms using a Galois lattice structure, Proceedings of the Third International Conference on Tools for Artificial Intelligence, San Jose, CA: IEEE Computer Society Press, 22-29.

[22] Kim, M., and Compton, P. (2001). Formal Concept Analysis for Domain-Specific Document Retrieval Systems. In Proceedings of the 14th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence (AI '01), Markus Stumptner, Dan Corbett, and Michael J. Brooks (Eds.). Springer-Verlag, London, UK, UK, 237-248.

[23] Kuznetsov, S., & Obiedkov S. 2002. Comparing Performance of Algorithms for Generating Concept Lattices, 14, Journal of Experimental and Theoretical Artificial Intelligence, Taylor & Francis, ISSN 0952–813X print/ISSN 1362–3079 online, pp.189–216.