A Web-based Browsing Mechanism Based on Conceptual Structures

Mihye Kim and Paul Compton

School of Computer Science and Engineering
University of New South Wales, Sydney, NSW 2052, Australia
{mihyek, compton}@cse.unsw.edu.au

Abstract. Domain-specific information retrieval normally depends on general search engines which make no use of domain knowledge and require a user to look at a linear display of loosely organised search results, or hand-crafted specialised systems with a better browsing interface but which are costly to build and maintain. We are developing an approach to domain specific information retrieval that makes much greater use of domain expert knowledge. As part of this we have developed a browsing mechanism based on the Formal Concept Analysis (FCA) techniques as well as standard information retrieval techniques. This has been implemented and is demonstrated with a test domain of papers presented at the Banff Knowledge Acquisition Workshops in recent years.

1 Introduction

The World Wide Web is taking over as the main means of providing information. Keeping pace with the evolution of the computer industry, organisations as well as individual users are putting information on the Web and using information retrieval systems to make it easier for users to find what they are looking for. However, the obvious problem with these general search mechanisms is setting up an appropriate query to find the particular documents that may be relevant to one’s particular interest.

The purpose of this study is to develop techniques for organisations as well as individuals to incrementally and easily build and maintain their own information retrieval systems. The particular objective in the work described here is to develop a suitable browsing mechanism for incrementally developed domain-specific document retrieval. This study extends earlier work where Ripple-Down Rules (RDR) was used for help desk information retrieval and which enable domain experts to constantly improve the system's retrieval (Kang et al. 1997; Kim et al. 1999). The RDR approach was initially developed for knowledge acquisition (KA) for knowledge based systems (Compton and Jansen 1990). The idea behind these systems is that when a user fails to find a suitable document the system sends the expert a log of the interaction. The RDR mechanism then assists the expert to add keywords and rules so that the correct document will be found next time. In the previous work we endeavoured to explore incremental knowledge acquisition and maintenance techniques of the system's knowledge by experts based on the RDR mechanism. In this paper the focus is on a browsing mechanism appropriate to RDR's incremental knowledge acquisition.

The browsing approach we propose uses Formal Concept Analysis (FCA). FCA has previously been used with RDR expert systems as an explanation tool (Richards and Compton 1997 & 1999). Here FCA is used for dynamic clustering and browsing for document retrieval. A hierarchical conceptual clustering of the documents is built dynamically with the results corresponding to the user's query. In browsing the concept lattice, we simplify the lattice display by showing only direct neighbours in the lattice using Hyperlinks rather than focusing on visualising the lattice graph itself. We also integrated the browsing with a standard query interface. Although the system supports incremental addition of documents and editing, this does not yet use the RDR approach to ensure a monotonic improvement in performance. This remains for future work. Our test environment is a system that gives access to all the Banff Knowledge Acquisition Workshop papers in recent years. Since 1996 all papers for this workshop have been published on the Web. Using the knowledge acquisition domain will also make possible comparisons with ontology based systems also applied to the knowledge acquisition domain (Benjamins et al, 1999)

In the next section, we will examine the use of FCA for information retrieval and outline the browsing mechanism we have developed. Then we will present the prototype we have implemented through examples. Finally we will outline future directions for this work.

2 FCA for Information Retrieval

In standard information retrieval, documents are indexed using normalised terms with term frequencies commonly based on a vector space or probabilistic model. In this approach users use a keyword search by formulating a Boolean query to find information or browse web directories which are organised as a hierarchy of subject categories. Another approach for information retrieval is ontology-based retrieval (KA^2; OIL; RDF; SHOE). In this approach, knowledge engineers or ontology developers first build ontologies for the domain and require that documents be annotated according to the ontologies in a similar fashion to traditional software engineering development. An ontological browser is supported as a query interface.

Another approach is based on a lattice-based retrieval formalised using FCA. A number of researchers have advanced this lattice structure for document retrieval (Godin et al. 1993; Carpineto and Romano 1995, 1996; Priss 1997, 2000). Several researchers have studied the concept lattice for domain-specific information retrieval (Cole and Eklund 1996; Cole and Stumme 2000; Eklund et al. 2000; TOSCANA). This is based on navigating a lattice structure as known a Galois lattice (Wille 1982) and more recently known as Formal Concept Analysis (Ganter and Wille 1999). Here we reviewed the literature focusing only on the cases of document retrieval as our implemented system is for document retrieval. In this case, documents are indexed based on the relationship of formal concepts. A key benefit of this method is in the lattice-browsing scheme, which has advantages over the hierarchical approach used in general search engines and ontology-based searches. Users can navigate the whole lattice by the generalising and specialising relationships between the formal concepts.

Godin et al. (1993) has addressed the advantage of the lattice method against hierarchical classification and also evaluated the retrieval performance compared to a conventional Boolean retrieval and the Galois lattice retrieval method. No significant performance difference was found for retrieval, but the lattice structure was suggested as being an attractive alternative because of the potential advantages of lattice browsing. Carpineto and Romano (1996) used a thesaurus as background knowledge to formulate browsing and presented experimental evidence that adding a thesaurus to a concept lattice improves its retrieval performance. The interface of Godin et al. (1993) was implemented on a standard screen for a Macintosh microcomputer using window, menu and dialog interface tools and viewing only direct neighbours in the lattice. The interface developed by Carpineto and Romano (1995) showed the lattice graph but a similar to a fisheye view (Furnas 1986) of individual nodes on a stand-alone Symbolic Lisp Machine. Both employed the controlled terms of a database to construct a lattice.

More recently FCA has been used to build a graphical interface for a document retrieval (Priss 1997) culminating in faceted information retrieval system (FaIR) (Priss 2000). It classifies the documents using a faceted knowledge representation based on a thesaurus or knowledge base. The prototype of FaIR has been implemented as an interface for a small knowledge base (http://kb.indiana.edu/), but a browsing scheme is not yet supported.

3 Browsing

The browsing mechanism we have developed is very similar to the approach of Godin et al. (1993) especially in dealing only with direct neighbour lattice nodes, rather than visualisation of the lattice graph. Much research has been devoted to viewing the concept lattice (Cole and Stumme 2000; Eklund 2000; Eklund et al. 2000; TOSCANA). Lin (1997) discussed how visualisation through a graphical interface could enhance information retrieval. However, we anticipate the viewing of the whole lattice dealing with large domains will remain a problem. We simplify the lattice display by showing only direct neighbours by hyperlinks, which provide a fairly natural simplification for a lattice display. We believe that the advantage of the lattice structure for information retrieval is retained with this interface. Another reason for this approach is that it is very natural for Web users.

In addition, we combine a Boolean query interface, which uses general search mechanisms and the FCA browsing interface. Our general search mechanism is similar the mode of general search engines on the Web. A hierarchical conceptual clustering of the documents is also built dynamically in response to the users' query. Users can use the two modes of dialogue (general search mechanism and lattice-based retrieval) in a systematic manner. Carpineto and Romano (1998) support a Boolean query interface to move to a portion of the concept lattice associated with the users' query. We also support the function when the users use the lattice-based retrieval. Furthermore, we support conceptual scaling. In the study here this is only used for the attribute of "authors", "proceedings title" and "publication year" to investigate its utility. We also incrementally construct the concept lattice by allowing new documents to be added and refinement of the set of concepts for existing documents.

Godin et al. (1993) and Carpineto and Romano (1995) both employed the controlled vocabulary from a database which already exists. We did not use controlled terms, rather we allow experts or authors to add both documents and a set of keywords into the system. We anticipate that the existence of the browsing scheme will help users chose appropriate keywords. When the browsing is integrated with RDR, the RDR system will assist the user in finding appropriate discriminating keywords. However a key contrast with previous systems is that we do not commit to either a predefined set of concepts on which clustering is developed and as used in large-scale search engines nor do we require the prior development of an ontology for the domain. Rather the experts have a more flexible role, adding concepts as the need arises. In our view the thesaurus (taxonomical ontology) should be discovered from the domain knowledge as documents are added, or when documents fail to be retrieved, by the authors or other domain expert adding concepts rather than simply incorporating and using a predefined ontology.

4 Construction of a Concept Lattice

Formal Concept Analysis (FCA) was developed by Wille (Wille 1982). The extension of a concept is formed by all objects to which the concept applies and the intension consists of all attributes existing in those objects. The set of objects and their attributes consist of a formal context. All formal concepts are found using given mathematical formulas from the formal context. Then the subconcept-superconcept relationships between concepts are expressed in a concept lattice. More detailed definitions and examples can be found in (Ganter and Wille 1999).

4.1 Finding Formal Concepts and Building a Concept Lattice

In our application documents correspond to objects and the keywords of the documents constitute attribute sets. Then we define a formal context (C) = (D, K, I) where D is a set of documents and K is a set of keywords and I is a binary relation which indicates where a document *d* has a keyword *k* by the relationship $(d, k) \in$ I. For example, Table 1 shows the formal context of C where D is {1, 2, 3, 4, 5}, K is {RDR, Knowledge acquisition, MCRDR, Simulated experts, Formal concept analysis, NRDR} and the relation I is {(1,RDR), (1, Knowledge acquisition), … , (5, RDR), (5, NRDR)}.

Table1. Part of formal context in our application

	RDR	Knowledge acquisition	MCRDR	Simulated experts	Formal concept analysis	NRDR
1	X	X				
2	X	X	X			
3	X		X	X		
4	X		X		X	
5	X					X

The following derivation is used to cultivate the formal concepts of a formal context. A formal concept is defined as a pair (X, Y) such that $X \subseteq D$, $Y \subseteq K$, $X' = Y$ and $Y' = X$ where X and Y are called the extent and the intent of the concept (X, Y).

$$X \subseteq D: \ X \mapsto X' := \{k \in K \ \mid \forall d \in X : (d, k) \in I\}$$
$$Y \subseteq K: \ Y \mapsto Y' := \{d \in D \mid \forall k \in Y : (d, k) \in I\}$$

The formal concepts of C are expressed in a concept lattice which is the basic conceptual structure of FCA and ordered by the smallest set of attributes as shown in Figure 1. To build a concept lattice we need to find the subconcept-superconcept relationship between the formal concepts. This is formalised by

$$(X_1, Y_1) \leq (X_2, Y_2) \Leftrightarrow X_1 \subseteq X_2 \ (\Leftrightarrow Y_2 \subseteq Y_1)$$

Where (X_1, Y_1) is called a subconcept of (X_2, Y_2)

(X_2, Y_2) is called a superconcept of (X_1, Y_1)

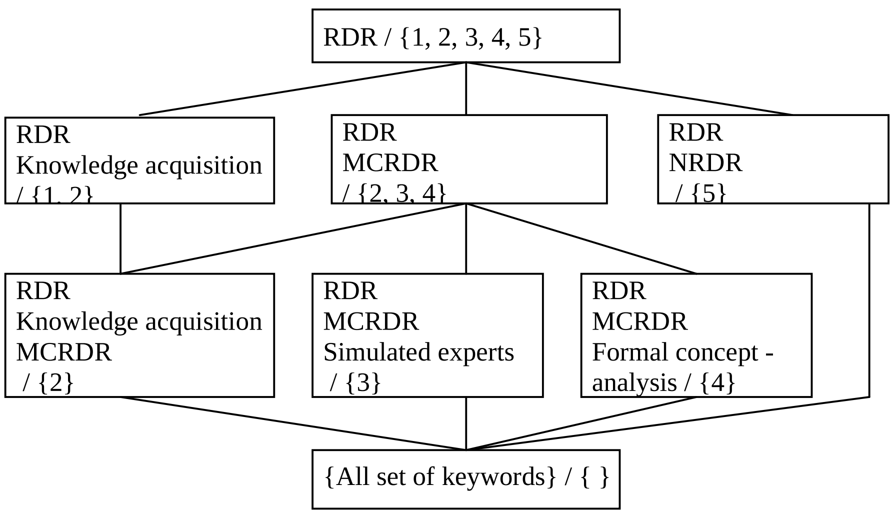

Fig.1. Concept lattice of the formal context in Table 1

4.2 Conceptual Scaling

Conceptual and relational scaling has been introduced in order to deal with many-valued attributes (Ganter and Wille 1989, Prediger and Wille 1999). Some sets of Boolean attributes may be more appropriately viewed as mutually exclusive values for a multi-valued attribute. A many-valued context is defined as a formal context (K) = (G, M, W, I) where G is a set of objects, M is a set of attributes, W is a set of attribute values. I is a ternary relation between G, M and W which indicates where an object *g* has the attributes value *w* for the attribute *m*. A many-valued context is then turned into a number of formal contexts. This process is called conceptual scaling. The concept lattices are structured for each of the separated formal contexts.

Table 2. An example table for many-valued contexts

	Concepts	Authors	Proceeding title	Publication year
Doc1	c1,c2, c3	a1,a2	KAW	1996
Doc2	c1, c3, c4	a3,a2	EKAW	1999
Doc3	c1, c2, c5, c6	a4, a5, a6	PKAW	2000
…	…	…	…	…

Then, a concept lattice is derived by combing several concept lattices into 'nested line diagrams' (TOSCANA) or a new form of a lattice structure. Relational scaling is the extension of conceptual scaling for deriving structures of concept graphs from many-valued contexts in addition to concept lattices (Prediger and Wille 1999). In this paper we deal only with the conceptual scaling.

Table 2 is an example of many-valued contexts in our prototype domain. We build a concept lattice with a set of documents with their keywords as an outer structure. Then, we scale up using other attributes ('author', 'proceeding title' and 'publication year') into a nested structure. The nested structure is constructed dynamically in response to the concept of the outer structure. In the example here we have selected obvious multi-valued attributes.

Conceptual scaling is also applied to one-valued contexts in order to reduce the complexity of the visualisation (Cole and Stumme 2000). In this case, it can be seen as a method to group together related values of the attribute context. In our system, the user can group attribute values from the one-valued contexts. The process is incorporated with building a taxonomical ontology. Then, conceptual scales which correspond to the user's query are derived from the ontology at run-time. In the final system, the user (or expert) will be able to group attributes as more appropriately being values for a multi-valued attribute on the fly.

4.3 Incremental Construction of the Concept Lattice

It is also essential to be able to incrementally construct a concept lattice by adding a new document and refining the existing information in the system. The set of objects (documents) can be added in a batch, but it is more likely that documents will be added individually. Godin et al (1991) and Carpineto and Romano (1993) proposed an incremental algorithm for updating the concept lattice. In experimental tests the incremental algorithms, even the simplest and least efficient, outperform all the batch algorithms (Godin et al 1995). This study also observed that the time for adding a new object is proportional to the number of objects on the average. We have developed a further incremental algorithm to construct the concept lattice. It may be appropriate to carry out a comparison of the incremental algorithms but this is beyond the scope of this paper.

In our approach, the formal context, formal concepts and lattice structure are incrementally changed by adding a new document with a set of keywords or by refining the keywords for an existing document. Assume the existing formal context (C) = (D, K, I) where D is a set of documents, K is a set of keywords and I is a binary relation between D and K. Then, £(C) be the set of all formal concepts of the formal context C. A formal concept of the context C consists of a pair (X, Y) where X and Y are called the extent and the intent respectively. Now let X′ be all extents and Y′ be all intents of £(C). In the case of adding a new document, let *d* be a new document and Γ be the set of keywords of *d*. Then, an extended formal context of C is defined as $C^+ = (D^+, K^+, I^+)$ where $D^+ = D \cup \{d\}$, $K^+ = K \cup \Gamma$ and $I^+ = I \cup \{(d, k) \mid k \in \Gamma\}$. In the case of refining an existing case, $D^+ = D$ and $K^+ = (K - \Gamma) \cup \Gamma'$ where Γ′ is the new case of keyword of the refining document.

Then the following procedure will be applied for each element (*k*) of Γ. The system formulates a formal concept (X, Y) where X is the set of documents which is associated with the element *k* and Y is {*k*}, and determines the intersection of X with the X′ of £(C). If the intersection does not exist in X′, the system reformulates the formal concept (X, Y) with X is the intersection and Y is {*k*} and adds the concept into £(C). After this process, the extended set of all formal concepts $£^+(C)$ is composed. But $£^+(C)$ can include a common attribute component contrary to FCA. We eliminate the formal concepts in the common attribute component except for the maximal concept of $£^+(C)$ defined with the largest object component. For reference, the object components of the common attribute component are in a total subsumption relationship. Then, subconcept and superconcept relationships are reformulated for all formal concepts which include the keywords Γ of *d*. This results in a new lattice $£(D^+, K^+, I^+)$ of the context C^+.

In our implementation, when a document is added or the set of keywords for a document is refined, the experts (or users) can directly view the changed concept lattice and can navigate the concepts they have added. After observing the represented relationships between the concepts, they can make a decision whether the set of keywords they have assigned for the document is appropriate or needs refinement.

5 Implementation

A prototype has been implemented on the World Wide Web (URL: http://pokey.cse.unsw.edu.au/servlets/Search). The knowledge base of the system is built with information extracted from Knowledge Acquisition Workshop Papers (KAW96, KAW98, KAW99; URL: http://ksi.cpsc.ucalgay.ca:80/ KAW/) with 200 documents. The system was developed with Java, JavaScript and Java Servlets (Java CGI) supported by a web browser such as Netscape 4.0 and Explorer 5.0 or higher.

5.1 Document Retrieval

A user can specify a query by selecting among a given set of concepts (keywords) for the domain shown in a pull-down menu or by entering any textwords in a conventional information retrieval fashion shown in Figure 2. A set of words can be entered separated by commas (',') assuming the AND Boolean operator. A set of words separated by spaces is treated as a phrase search. We use 'stopwords' and strict 'stemming' classes. The entire 5.5 gigabyte TREC 1 - 5 collection was used to create the stemming classes by merging Porter and K-Stem stemming algorithms which has given the overall best result on the TREC6 experiments. The purpose of the stemming is to deal mainly with the plural problem, rather than sophisticated morphological processing. We also added classes related to KAW papers. However, at this stage we have made little effort to provide appropriate classes. The advantage of this is that there will be plenty of scope for further development using RDR knowledge acquisition.

When a query is entered, stopwords are first eliminated and the remaining query stemmed using the stemming classes. Next, the system decides whether the query exists in the set of keywords. If the query exists in the keywords, the documents that include the keywords are retrieved. If not, the system tries to identify documents from the document context. That is, the keywords are treated as textwords that may occur in the document. Then the documents which match the query are retrieved. In the latter case, the system sends a log file to the experts so they can refine the keywords for the documents. After these procedures, the system formulates a sub-concept lattice with the set of objects D′ from the retrieved documents and the set of attributes K′ with the keywords in all the documents of D′. Then, a hierarchical conceptual clustering for each formal concept at the first level of the lattice is built using their child formal concepts for the three levels. A browsing structure is also constructed based on the concept lattice. Figure 2 shows the retrieved documents and the clustering.

The bottom-right of the screen shows the results. By clicking on the Hyperlinks, the user can see the whole content of the documents as commonly used in WWW pages. The bottom left-hand side of the screen shows the clustering which indicates parent-child relationships in a hierarchy. Through it, the user can explore the domain knowledge quickly and refine his/her query easily. The clustering interface is implemented using a pop-up menu-driven technique shown in Figure 2. More specific documents can be obtained by clicking one of the menu items. Then, the result will be changed to the documents which satisfy all clicked items. The number displayed with the clusters indicates the number of documents in the cluster. The user can continue his/her search work by refining the previous query or by formulating a new query. The clustering interface may not be necessary as the browsing interface provides the same functionality. But, at this stage we have left the system with both a more conventional hierarchical interface and a browsing interface.

5.2 Navigating the Concept Lattice

A user can navigate the relevant documents based on the lattice structure by clicking the hyperlink of "Navigating" in Figure 2. If the user clicks the hyperlink, the full screen of the new window will pop up as shown in Figure 3. As described in the previous section, if the query exists in the set of keywords, the documents that include the keywords are retrieved. If not, the system tries to find the keywords in the text of the documents. Then, the documents which satisfy the query are retrieved. After that, the system formulates a sub-concept lattice dynamically with the set of objects D′ from the retrieved documents and the set of attributes K′ from the keywords in all the documents of D′ and moves to a relevant portion of the lattice.

All direct parent and child concepts are displayed. The user can easily discover which concepts are in relationship to each other. To facilitate a user's understanding

Fig. 2. Search result and clustering

of parent and child concepts, we use different colours (red for parents, green for the current concepts and blue for the child concepts) and display the parents' concepts first and display concepts alphabetically. The user can navigate the parents' concepts to get more general documents and navigate the child concepts to get more specific documents. If the user clicks a hyperlink (a concept), the screen will be changed with the parent and child concepts of the hyperlink. The search results are obtained by clicking the icon ( ) existing in the front of each concept. The user interface using the hyperlinks seems to us not only simpler, but also more useable than drawing the lattice graph itself. The user can search documents by other attributes by specifying a value or selecting from the pull-down menu for each attribute. The set of attribute values (here author, proceeding title and year) are automatically accumulated by submitting a paper to the system. The browsing for the results is performed in the same way described in the above.

At this stage we have also included a simple tool for experts to build taxonomical ontologies ("Maintain KB" hyperlink in Figure3) to extend the user's query and link formal concepts. Based on the ontology and synonym classes, we suggest 'concepts' which have relationships *(is-a or synonym)* with the query. We assume that a virtual connection is formed between lattice nodes of the concept lattice based on the ontology. Through the virtual connection (or a virtual lattice), the user can directly jump from one lattice node to another lattice node. It is required a better technique to combine the concept lattice with ontologies. At this stage this is very much a prototype to see if such a facility improves browsing. This sort of ontology development will also be integrated into the RDR knowledge acquisition. In the longer term it would be appropriate to import ontologies, for example the $(KA)^2$ initiative ontologies.

Fig. 3. An example of navigating the concept lattice

5.3 Browsing the Nested Structure

The user can also narrow the results using the nested attributes by clicking on the 'nested' icon existing in the front of the current formal concept as shown in Figure 4. We build a concept lattice using the result documents with their keywords as an outer structure and scale up with other attributes into a nested structure. The nested structure is constructed dynamically and associated with the current concept of the outer structure. That is, the nested attribute values are extracted from the result documents. The interface is implemented using a pop-up menu-driven technique shown in Figure 5, rather than a line diagram. But we may need to adapt a better user interface to cover many multi-valued attributes. If the user clicks on one of the menu items, the results will be changed according to the selection. The user can navigate recursively among the nested attributes.

In the current implementation, the attributes used for the nested are pre-decided (author, publication title and year). The user can group attributes values in the one-valued contexts (here a set of keywords) while building taxonomical ontologies. Then, nested attributes for that are associated with the user's query are derived from the ontology. The 'RDR' which exists in the nested menu in Figure 4 is an example of this case.

However, we intend that the user can group attributes on the fly. At the expert's discretion these could then be permanently added to the system. We would hope in the longer term that the system could suggest candidate attributes for consideration as values of a multi-valued attribute.

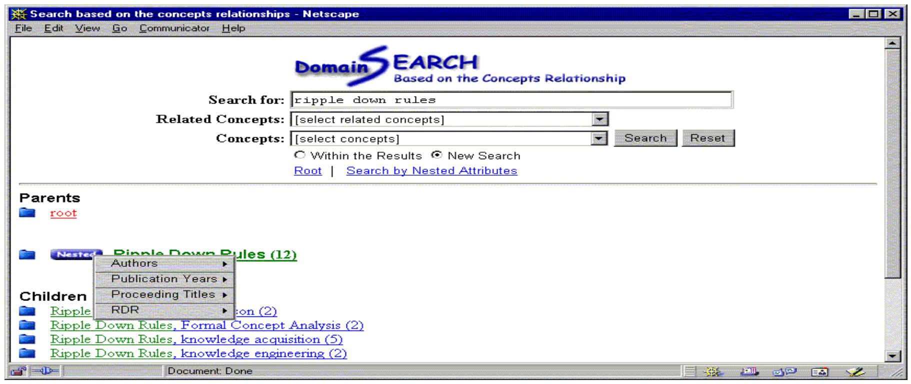

Fig. 4. An example of the nested attributes

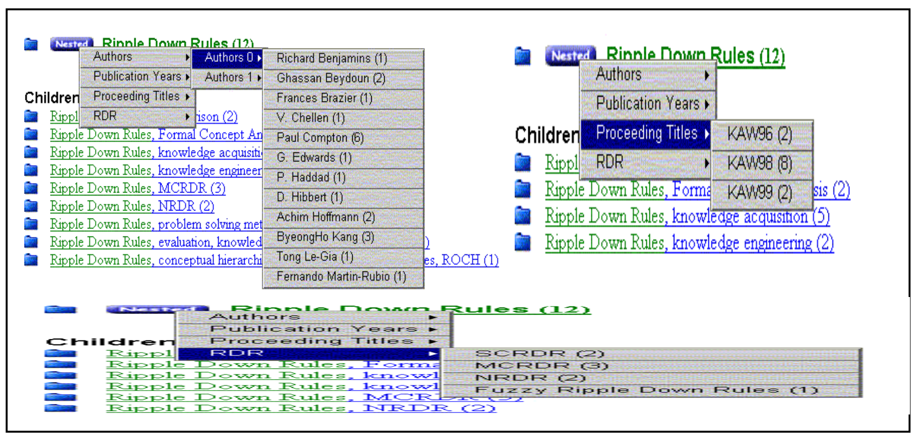

Fig. 5. Snippet browsing examples of the nested attributes

5.4 Adding a Document and Refining Concepts

Experts can add a document to the system by entering a set of attribute values. The whole concept lattice will be incrementally changed. Then, experts can observe the relationship of the concepts in the lattice along with the existing documents to decide whether the keywords used are appropriate and they can refine the concepts of the document. The concept lattice is also reconstructed incrementally to cover the change.

As noted a number of times already, we have already developed an RDR knowledge acquisition technique for documents and this will be integrated with the FCA browsing proposed here. It should be noted that FCA is also used for a knowledge acquisition process (Erdmann 1998; Wille 1992) to discover concepts and rules related to the objects and their attributes. This approach is based on a strong idea of context with its use of parent child-relations between concepts. However, the general principle in this work is to give the expert a view of the whole domain so that all relevant concepts will be included. In contrast, we have argued that experts more easily provide concepts that distinguish between cases (Compton and Jansen 1990). The expert's attention is focussed on relevant cases by the system misapplying a concept to case. The expert is then asked to distinguish between this case and a case the system retrieves where the concept was appropriate. This is a more strongly situated view of knowledge acquisition with more emphasis on the significance of context (Compton and Jansen 1990; Richards and Compton 1998; Tecuci 1998). It also relates closely to the repertory grid approach where the expert is asked to provide axes of differentiation (constructs) between cases (Gaines and Shaw 1990).

6 Summary

This study has taken a step in the direction of finding an approach to a Web-based document retrieval system suitable for specialised domains. This approach uses standard information retrieval techniques but also emphasises the role of domain experts in building such systems. In previous work we have explored the use of RDR to support experts in adding keywords or concepts to documents in response to query failure so that the system is better adapted to a user's natural way of formulating queries. Here we have demonstrated a browsing mechanism that will enable users to more easily explore documents appropriate to specialised domains and based on the involvement of experts in assigning concepts to documents. We speculate that there has been little follow up of earlier work using FCA for information retrieval as it is likely the techniques become unwieldy when developed as a general IR technique based on all the words in a document. We see the method as applying only to fairly small sets of keywords attached to documents by experts. However, we provide a conventional use of text words to find some documents that are then used to form a concept lattice based on the expert-added concepts.

Although FCA and RDR seem an attractive solution to specialised domain information retrieval, there are also a number of research issues to be explored. First of all, we need to integrate the FCA browsing and RDR knowledge acquisition techniques to facilitate incremental knowledge acquisition mechanisms. The next step is to evaluate this approach in routine use with reasonably large data sets, but still related to specific domains. Another related issue is that we do not support controlled vocabularies for the domain in advance. Although experts are involved we do not commit to either a predefined set of concepts on which clustering is developed and as used in large-scale search engines nor do we require the prior development of an ontology for the domain. However a major area where this work needs to expand is in providing support for experts so that they re-use keywords and that they can develop taxonomies if required.

At a more fundamental level, the value of FCA for IR is based on the assumption that when you enter a keyword, but the documents retrieved are inappropriate, then these documents will have other keywords that will eventually lead you to the desired documents. This is a central but hidden assumption in proposing that a lattice-browsing scheme will have advantages over a hierarchical approach. In a hierarchical scheme you simply go back to the top and start again. With a lattice approach you assume that there are other features of the retrieved document that will also occur in the documents you really want to retrieve. This is a central and critical assumption that needs to be explored further.

In summary, we have not yet fully developed and evaluated this form of expert-centred information retrieval. However, this prototype at least suggests the possibility of a new way of information retrieval where an expert can rapidly build and maintain an information retrieval system in his or her area of expertise which will be easy for domain users. We believe that these highly specialised, 'disposable' systems will be critical in making full use of the enormous amounts of knowledge appearing in Intranets and the Internet itself.

References

1. Benjamins, V. R., Fensel, D., Decker, S. and Perez, A. G. (1999). $(KA)^2$: building ontologies for the Internet: a mid-term report, *International journal of human computer studies*, Vol. 51, No. 3, 687-712.
2. Carpineto, C. and Romano, G. (1993). GALOIS: An Order-Theoretic Approach to Conceptual Clustering, *In Proceedings of the Machine Learning Conference*, 33-40.
3. Carpineto, C. and Romano, G. (1995). ULYSSES: A Lattice-based Multiple Interaction Strategy Retrieval Interface, In Blumenthal et al., *Human-Computer Interaction*, Springer Verlag.
4. Carpineto, Carpineto, C. and Romano, G. (1996). A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval, *Machine Learning*, 24(2), 95-122.
5. Carpineto, Carpineto, C. and Romano, G. (1998). Effective reformulation of boolean queries with concept lattices, *Proceedings of the Third International Conference on Flexible Query Answering Systems*, Roskilde, DK, 83-94.
6. Cole, R. and Eklund, P. (1996). The Application of Formal Concept Analysis to Text Retrieval in the Medical Domain, *In the Proceedings of the 4th International Conference on Conceptual Structures.*
7. Cole, R. and Stumme, G. (2000). CEM - A Conceptual Email Manager, *In the Proceedings of the 8th International Conference on Conceptual Structure ICCS 2000*, Darmstadt, Springer Verlag, 438-452.
8. Compton, P. and Jansen, R. (1990). A Philosophical Basis for Knowledge Acquisition, *Knowledge Acquisition* 2:241-257.
9. Eklund, P. (2000). Logic-Based Networks: Concept Graphs and Conceptual Structures, *In the Proceedings of the 8th International Conference on Conceptual Structure ICCS 2000*, Darmstadt, Springer Verlag, 399-420.
10. Eklund, P., Groh, B., Stumme, G. and Wille, R. (2000). A Contextual-Logic Extension of TOSCANA, *In the Proceedings of the 8th International Conference on Conceptual Structure ICCS 2000*, Darmstadt, Springer Verlag, 453-467.
11. Erdmann, E. (1998). Formal Concept Analysis to Learn from the Sisyphus-III Material, *Eleventh Workshop on Knowledge Acquisition, Modeling and Management (KAW'98)*, Banff, Alberta, Canada.
12. Furnas, G. W. (1986). Generalized fisheye views, *Proceedings of the Human Factors in Computing Systems*, North Holland, 16-23.
13. Gaines, B. and Shaw, M. (1990). Cognitive and Logical Foundation of Knowledge Acquisition, *The 5th Knowledge Acquisition for Knowledge Based Systems Workshop*, Banff, 9.1-9.25.

14. 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.
15. Ganter, B. and Wille, R. (1999). Formal Concept Analysis: mathematical foundations, Springer, Heidelberg.
16. 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.
17. Godin, R., Missaoui, R. and April, A. (1993). Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods, *International Journal of Man-Machine Studies*, 38:747-767.
18. Godin, R., Missaoui, R. and Alaoui, H. (1995). Incremental concept formulation algorithms based on Galois (concept) lattices, *Computational Intelligence*, 11(2), 246-267.
19. Kang, B. H., Yoshida, K., Motoda, H. and Compton, P. (1997). Help Desk System with Intelligent Interface, *Applied Artificial Intelligence*, 11: 611-631.
20. Kim, M., Compton, P. and Kang, B. H. (1999). Incremental Development of a Web Based Help Desk System, *Proceedings of the 4th Australian Knowledge Acquisition Workshop (AKAW99)*, University of NSW, Sydney, 13-29.
21. Lin, X. (1997). Map Displays for Information Retrieval, *Journal of the American Society of Information Science*, 48:40-54.
22. Luke, S., Spector, L., Rager, D. and Hendler, J. (1997). Ontology-based knowledge discovery on the world-wide web, Proceedings *of the Workshop on Internet-based Information Systems, AAAI'96*, http://www.cs.umd.edu/ projects/plus/SHOE/.
23. OIL. (2001). Ontology Inference Layer, http://www.ontoknowledge.org/oil.
24. Prediger, S. and Wille, R. (1999). The lattice of concept graphs of a relationally scaled context. In: W. Tepfenhart, W. Cyre (eds.): *Conceptual Structures: Standards and Practices*, LNAI 1640, Springer, Heidelberg.
25. Priss, U. (1997). A Graphical Interface for Document Retrieval Based on Formal Concept Analysis, *Proceedings of the 8th Midwest Artificial Intelligence and Cognitive Science Conference*, AAAI Technical Report CF-97-01, 66-70.
26. Priss, U. (2000). Faceted Information Representation, In: Stumme, Gerd (ed.), Working with Conceptual Structures. *Proceedings of the 8th International Conference on Conceptual Structures*, Shaker Verlag, Aachen, 84-94.
27. RDF. (2001). Resource description framework, http://www.w3 .org/RDF, 2001.
28. Richards, D. and Compton, P. (1997). Knowledge acquisition first, modelling later, *Knowledge Acquisition, Modeling and Management*, E. Plaza and R. Benjamins, Berlin, Springer: 237-252.
29. Richards, D. and Compton, P. (1998). Taking up the Situated Cognition Challenge With Ripple Down Rules, *International Journal of Human-Computer Studies* 49:895-926.
30. Richards, D. and Compton, P. (1999). "An alternative verification and validation technique for an alternative knowledge representation and acquisition technique", *Knowledge-Based Systems* 12: 55-73.
31. Tecuci, G. (1998). Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies, Sydney, Academic Press.
32. Wille, R. (1982). Restructuring lattice theory: an approach based on hierarchies of concepts, In: Ivan Rival (ed.), Ordered sets, Reidel, *Dordrecht-Boston*, 445-470.
33. Wille, R. (1992). Concept lattices and conceptual knowledge systems, *Computers and Mathematics with Applications*, 23:493-515.