Incremental Development of Domain-Specific Document Retrieval Systems

Mihye Kim and Paul Compton
School of Computer Science and Engineering
University of New South Wales
Sydney NSW 2052 Australia
+61 2 9385 6531
{mihyek, compton}@cse.unsw.edu.au

ABSTRACT

We have proposed and implemented a browsing mechanism for domain-specific document retrieval systems based on the conceptual structure of Formal Concept Analysis. This paper concerns how knowledge acquisition mechanisms for incremental development are activated to improve the system's quality as browsing evolves. The knowledge acquisition mechanisms for the system are based on Formal Concept Analysis (FCA) and Ripple-Down Rule (RDR) knowledge acquisition techniques. Our experience with the prototype suggests that a user can fairly easily add and annotate new documents so that they can be readily retrieved and also distinguished from less relevant documents. It also appears that our approach can apply to conceptual modelling of domain taxonomies collaboratively created and maintained over time by users (or authors) without the mediation of knowledge engineers and suggest how the resultant ontology may be integrated with other ontologies.

INTRODUCTION

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 handcrafted specialised systems with a better browsing interface but which are costly to build and maintain.

An alternative approach for specialized domains can be to allow users to create their own organization of documents and to assist them in ensuring improvement of the system's performance as it evolves. The purpose of this study is to develop such a system suitable for organisations as well as individuals to incrementally and easily build and maintain their own information retrieval systems. We have proposed and implemented a browsing mechanism for such a system based on the conceptual structure of Formal Concept Analysis [16, 17]. This paper concerns how knowledge acquisition mechanisms for incremental development are used to improve the system's quality as browsing evolves.

We have previously demonstrated incremental development of document management systems based on selecting keywords that discriminate between documents [14, 15] using RDR knowledge acquisition and maintenance methodology. The RDR approach was initially developed for knowledge acquisition for knowledge based systems [7]. Although, as demonstrated in other RDR work, RDR greatly assists context-specific knowledge acquisition. it does not organise the knowledge in a way that is suitable for browsing. One of the aims here is to integrate the RDR incremental approach with the browsing advantages of FCA. FCA has previously been used with RDR expert systems as an explanation tool [21].

Our test environment is a system (http://pokey.cse.unsw.edu.au/servlets/Search) that gives access to all the Banff Knowledge Acquisition Workshop papers with around 200 in recent years (http://ksi.cpsc.ucalgay.ca:80/KAW). Another test domain (http://pokey.cse.unsw.edu.au/ servlets /RI) is for research topics in the School of Computer Science and Engineering, UNSW. There are around 150 research staff and students in the School who generally have homepages indicating their research projects. The aim here was to allow staff and students to freely annotate their pages so that they would be found appropriately within the evolving lattice of research topics. The goal is a system to assist prospective students and potential collaborators in finding research relevant to their interests.

In the next section, we address our approach by reviewing related approaches in literature. Next we describe FCA to explain how lattice-based browsing is formed and how knowledge acquisition mechanisms for incremental development are activated. Then we present the system that we have implemented focusing on knowledge acquisition strategies. Finally we discuss how this could be further improved particularly in how it may be integrated with other ontologies.

SYSTEM APPROACH

Broadly speaking there are two ways in which a user interacts with document retrieval systems. In one the user formulates a specific query and some documents are retrieved in response. This process is normally iterative in that the user refines (or changes) the query on the basis of the documents retrieved by each query. In the second approach the documents are grouped and the document groups organised into some sort of structure that can be browsed. That is, from any point in the structure at least some other related parts of the structure can be identified and moved to.

The ideal would be that specific queries would always produce the most relevant documents. Despite improvements in this area (e.g. Google), specific queries remain very frustrating: the only search terms the user can think of, occur in myriad other contexts and perhaps even do not occur in some relevant documents. As a result a browsing approach is supported in many information retrieval systems. With browsing users quickly explore the search domains and can easily acquire domain knowledge [19]. Typically, a hierarchy is used for browsing and documents are grouped using some sort of clustering algorithms. Hierarchical Agglomerative Clustering (HAC) algorithms are probably the most commonly used.

The problem with a hierarchical clustering is category mismatch [9, 13]. If one goes down the wrong path one must go back up the hierarchy and start again. There is no mechanism for navigating to other clusters, as there is only a simple taxonomy structure. A further critical issue in a browsing scheme is the origin of the terms by which the documents are grouped. One can attempt to arrive at some global taxonomy to satisfy all possible users as used by sites like Yahoo and the Open Directory Project (http://dmoz.org/). In these global systems the category mismatch problems can be very severe.

As an alternative, documents can be organised using ontologies for browsing a specific domain. In recent years ontologies have become a major subject of interest in applications ranging from search using large taxonomies categorising Web sites in general (such as the Open Directory Project) or specific communities (such as KA^2) to e-commerce (such as on amazon.com). One of main aims of this approach is to facilitate the sharing of information between communities as well as individuals within the groups. An ontology infrastructure for the Semantic Web is now also under way along with Web-based ontology representation languages such as XML/S (http://www.w3.org/XML), RDF/S (http://www.w3.org/RDF), OIL (http://www.ontoknowledge.org/oil), DAML and DAML+OIL (http://www.daml.org). A good example of this type of activity is the (KA)^2 initiative [2]. (KA)^2 starts out with an ontology appropriate to the domain with the expectation that people in the community will annotate documents according to the ontology. These same users should also be able to use the ontology to retrieve documents entered by others. There are likely to be considerable practical advantages to even very large communities committing to specific ontologies, and part of education would be to learn these ontologies. However, despite the practical advantages of a community committing to ontologies, we have long held the view that at base any knowledge structure is a construct which should be allowed to evolve over time [7].

Hence rather than committing to *a priori* ontologies and expecting that all documents will be annotated according to the ontologies, our aim is to explore the possibilities of a system where the user can annotate a document however they like and that the ontologies will evolve accordingly. Rather than this being totally *ad hoc*, we would like the system to assist the user to make extensions to the ontologies that are in some way improvements. We are not concerned with automated or semi-automated ways of discovering an ontology appropriate to a document or corpus [1, 18]. Despite the potential of such approaches, from our more deconstructionist perspective, we are more interested in the role of the reader or user interpreting documents and deciding on their annotation and development of an ontology. The user here may be the individual user, an expert for a specialised domain or a small community. However, this does not preclude the inclusion of ontologies either constructed by an expert or an ontology imported from elsewhere, as part of the ontological structure preferred by the user.

An alternative to a hierarchy for browsing is lattice-based navigation using Formal Concept Analysis (FCA). In this approach, a document is annotated by an expert with a set of controlled terms. From this a concept lattice is constructed using the mathematical formulae of FCA. The significant advantage of this approach is that the mathematical formulae produce a conceptual structure which automatically provides generalisation and specialisation relationships among the concept nodes. This lattice structure allows one to reach a group of documents via one path, but then rather than going back up the same hierarchy and guessing another starting point, one can go to one of the other parents of the present node as a way of navigating across the domain.

A number of researchers have proposed this lattice structure for document retrieval [3, 4, 13, 20]. Several researchers have studied the concept lattice for domain-specific information retrieval [5, 6, 8, 22]. We have also developed an FCA-based browsing mechanism [16, 17]. The focus in our previous work was to examine the advantages and capabilities of the lattice-based retrieval. The main difference in the work here is a greater emphasis on incremental development and evolution, and knowledge acquisition tools to support this for specialised domains. Our aim is a browsing scheme which can be collaboratively created and maintained and where users evolve their own organisation of documents but are assisted in this to facilitate improvement of the system's performance as it evolves.

Another difference is that our focus is on a web-based system using a hypertext representation of the links to a node, but without a graphical display of the overall lattice.

FORMAL CONCEPT ANALYSIS FOR THE SYSTEM

Formal Concept Analysis (FCA) is a mathematical theory which formulates the understanding of a 'concept' as a unit of thought comprising its extension and intension as a way of modelling a domain [12, 24]. 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. FCA generates a conceptual hierarchy of the domain by finding all possible formal concepts which reflect the relationships between attributes and objects. The resulting subconcept-superconcept relationships between formal concepts are expressed in a concept lattice which can be seen as a semantic net providing "hierarchical conceptual clustering of the objects… and a representation of all implications between the attributes" [25]. More detailed definitions and examples can be found in [12].

Formal Contexts and Formal Concepts

The most basic data structure of FCA is a formal context. The set of objects and their attributes constitute a formal context (K) = (G, M, I). G is a set of objects, M is a set of attributes and I is a binary relation between G and M which indicates where an object *g* has an attribute *m* by the relationship *gIm* (also by $(g, m) \in I$). In the original formulation of FCA, objects were implicitly assumed to have some sort of unity or identity so that the attributes applied to the whole object; e.g. a dog has four legs. Clearly documents do not have the sort of unity where attributes will necessarily apply to the whole document. However at this stage of this work we suppose that documents correspond to objects and the keywords or terms attached to documents by a user constitute attribute sets. We define a formal context (C) as follows for our document retrieval system.

Difinition1[1]: A formal context is a triple C = (D, K, I) where D is a set of documents, 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 ${}_dI_k$ (also by $(d, k) \in I$).

For example, Table 1 shows the formal context of C where D is {1, 2, 3, 4}, K is {artificial intelligence, information retrieval, machine learning, decision tree, natural language processing, speech recognition, signal representation} and the relation I is {(1, artificial intelligence), (1, information retrieval),..., (4, artificial intelligence), (4, natural language process), (4,speech recognition), (4, signal representation)}.

Then, formal concepts are derived from the formal context using the basic definition $X \subseteq D$: $X \mapsto X' := \{k \in K \mid \forall d \in X: (d, k) \in I\}$, $Y \subseteq K$: $Y \mapsto Y' := \{d \in D \mid \forall k \in Y: (d, k) \in I\}$. 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). More detailed mathematical formulae and procedures can be found in [12, 16, 24]. A node in Figure 1 represents a formal concept.

Concept Lattice

The formal concepts of C are expressed in a concept lattice £ (D, K, I) which is the conceptual structure of FCA and ordered by the smallest set of attributes. It is a basic

	Artificial Intelligence	Information Retrieval	Machine Learning	Decision Tree	Natural Language Processing	Speech Recognition	Signal Representation
Document 1	X	X					
Document 2	X		X	X			
Document 3	X	X			X		
Document 4	X				X	X	X

Table 1. Part of formal context in our application.

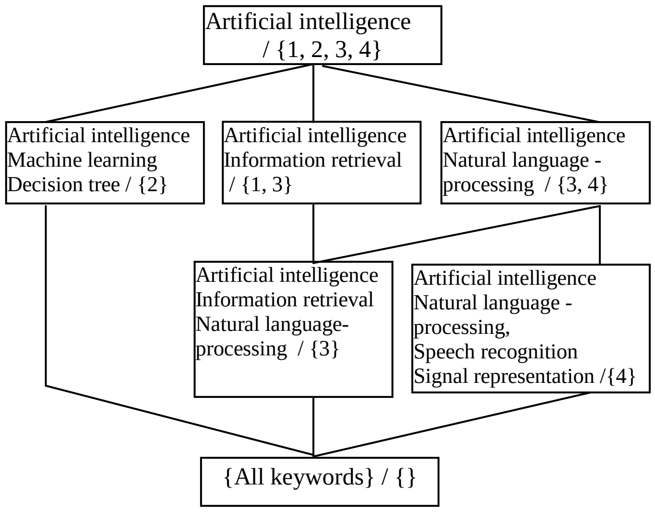

Artificial intelligence / {1, 2, 3, 4}
Artificial intelligence Machine learning Decision tree / {2}
Artificial intelligence Information retrieval / {1, 3}
Artificial intelligence Natural language - processing / {3, 4}
Artificial intelligence Information retrieval Natural language- processing / {3}
Artificial intelligence Natural language - processing, Speech recognition Signal representation /{4}
{All keywords} / {}

Figure 1. Concept lattice of the formal context in Table1.

structure for browsing in our approach. The structure is reformulated incrementally and automatically by adding a new case and refining the existing cases. 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) and (X_2, Y_2) is called a superconcept of (X_1, Y_1). Figure 1 shows the concept lattice of the formal context C in Table 1. This is an implicit and explicit representation of the data allowing a meaningful and comprehensible interpretation of the information seen as a sematic net [25]. In our approach the lattice is incrementally changed by adding a new case and refining the existing cases.

Conceptual Scaling

Conceptual scaling has been introduced in order to deal with many-valued attributes [11]. 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*. Then, if a user is interested in analysing the interrelationship between attributes, he/she can choose the required attribute(s) from the multi-valued context and build a formal context for the attribute(s). This process is called conceptual scaling. The concept lattices are structured for each of the separate formal contexts.

[1] This definition and the following material closely adhere to the Basic Theorem of FCA [24].

	Keywords	Authors	Proceeding title	Publication year
doc1	k1,k2,k3	a1,a2	KAW	1996
doc2	k1,k3,k4	a3,a2	EKAW	1999
doc3	k1,k2,k5	a4,a5,a6	PKAW	2000
…	…	…	…	…

Table 2. An example table for many-valued contexts.

A concept lattice is derived by combing several concept lattices into 'nested line diagrams' (e.g. TOSCANA) or a new form of a lattice structure. Table 2 is an example of many-valued contexts in our 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 into a nested structure. The nested structure is constructed dynamically in response to the outer structure. Conceptual scaling is also applied to one-valued contexts in order to reduce the complexity of the visualisation [23]. In our present system, an expert can group relevant attribute values from the formal context C = (D, K, I) in the definition 1. The process is incorporated with building a thesaurus which is not addressed here as it is beyond the scope of this paper. Then, when a query is associated with the thesaurus, conceptual scales are derived on the fly to group the relevant terms as the nested attributes.

Incremental Knowledge Acquisition

Knowledge acquisition is carried out when a new document is added with a set of keywords or the keywords of existing documents are refined. When an expert/user assigns the set of keywords for a document, some keywords may be prompted in regard to stored documents or domain knowledge. The purpose of incremental knowledge acquisition is to be able to obtain the concepts which are missed or unknown when concepts are first assigned for a document. The system guides the user to discover possible missed concepts through a number of steps. The knowledge acquisition mechanisms are based on FCA and RDR techniques. The following definitions are used.

Definition 2: Let C = (D, K, I) be a formal context, and d be a new document ($d \notin$ D) and Γ be the set of keywords of d. The set of keywords is not necessarily a subset of K. Then, the extended formal context of C is defined as $C^+ = (D^+, K^+, I^+)$ where $D^+ = D \cup \{d\}$, $K^+ = K \cup \Gamma$ and $I^+ = I \cup \{(d, k) \mid k \in \Gamma\}$.

Definition 3: Let C = (D, K, I) be a formal context and Γ be a set of keywords ($\Gamma \subseteq K$). Then the set of documents associated with Γ is defined to be $\Delta_\Gamma = \{d \in D \mid \exists k \in \Gamma$ such that $(d, k) \in I\}$.

We introduce Δ_Γ to get a set of documents which has at least one keyword of Γ. If Γ is a singleton (i.e. $\Gamma = \{\gamma\}$), then we will abbreviate $\Delta\gamma = \Delta_\Gamma = \{d \in D \mid (d, \gamma) \in I\}$.

Definition 4: Let C = (D, K, I) be a formal context. We define a function f from D to 2^K as f: D→ 2^K such that $f(d) = \{k \in K \mid (d, k) \in I\}$.

That is, $f(d)$ returns the set of keywords of d. Let the new document be d ($\notin$ D) with the set of keywords Γ. We formulate the sub-formal context C′ = (D′, K′, I′) with $D' = \Delta_\Gamma + \{d\}$ where Δ_Γ is in definition 3 and $K' = \bigcup_{d \in D'} f(d)$ where f is the function in definition 4. In order to get a set of relevant keywords of d, we obtain a set of keywords which are associated with Δ_Γ as $f(\Delta_\Gamma) = \bigcup_{d \in \Delta_r} f(d)$ from the context C′. Now the set of relevant keywords is defined as $\Re = f(\Delta_\Gamma) - \Gamma$. Then, the function *Freq* introduced below is used for each keyword of $\Re$ (k) to compute the number of common keywords of Γ with the keywords of all the documents that have the keyword k from the context C′.

Definition 5: We define a function *Freq* from $2^K \times K$ to the set of natural numbers N as follows: *Freq*: $2^K \times K \rightarrow N$ such that $Freq(\Gamma, k) = \sum_{d \in \Delta_k} |f(d) \cap \Gamma|$ where |X| is the cardinality of X.

The user can annotate his/her document with a set of keywords by entering any terms or selecting known terms. The system displays all the keywords used by other annotators to be able to share and reuse them. After this initial assignment, the user can view the other terms that co-occur with the terms s/he has provided and can annotate the document with these further terms if desired. The terms are presented to the user ordered by their frequency in the lattice, normalised for the number of terms at the node, and their 'closeness' to the node to which the document is assigned by the user's initial choice of terms in the conceptual hierarchy.

In a more detailed explanation, an ordered set of documents and a set of keywords which are relevant to the new document are obtained. A sub-lattice £′ (D′, K′, I′) of the formal context C′ described above is then constructed. This step is divided into two stages. In the first stage, the ordered documents are shown to the user along with the features that are different between the new document and each of the set of documents. Given a new document d, we are interested in finding the set of documents D_d that share some commonalties. We formulate a formal concept ζ ($\{d\}, f(d)$) with the newly added document d and its set of keywords Γ. Starting from the concept ζ we recursively go up to the direct superconcepts of its subconcept in the lattice to find the next level of the relevant documents. This procedure is done until the superconcept reaches the top node of the lattice.

For instance, suppose that there is a concept lattice as shown in Figure 1 and, a new document d (5) is added together with its set of keywords Γ {natural language processing, speech recognition, verbal interference}. Then, we formulate the sub-context C′ = (D′, K′, I′) where $D' = \Delta_\Gamma + \{d\} = \{3, 4, 5\}$, $K' = \bigcup_{d \in D'} f(d)$ = {artificial intelligence, information retrieval, natural language processing, speech recognition, signal representation, verbal interference} and I′ is a binary relation between D′ and K′. The sub-lattice

£(D′, K′, I′) of the context C′ can be constructed as shown in Figure 2. The grey coloured box indicates the formal concept ζ. From the lattice we can get the document *'4'* as it exists in the direct superconcept of ζ in the lattice and as such is the most relevant to the document *'5'*. Next the document *'3'* is obtained. Finally, we get an ordered set of documents {4, 3} relevant to the document *'5'* in the lattice. The ordered documents are then suggested to the user along with the features that differ between the new document and each of the relevant documents.

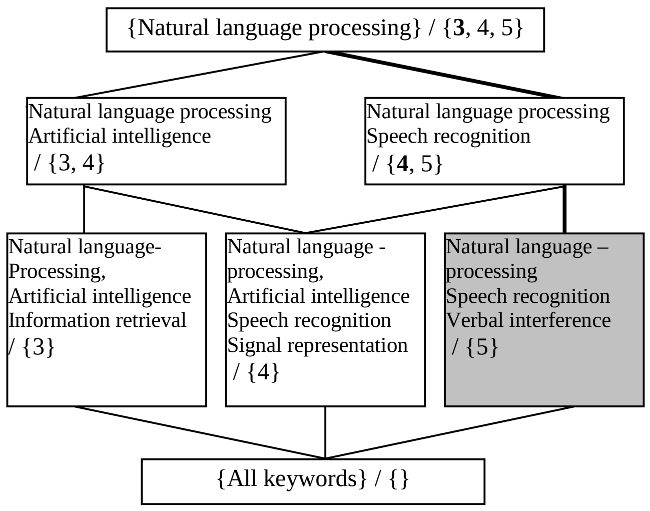

{Natural language processing} / {**3**, 4, 5}
Natural language processing Artificial intelligence / {3, 4}
Natural language processing Speech recognition / {**4**, 5}
Natural language-Processing, Artificial intelligence Information retrieval / {3}
Natural language - processing, Artificial intelligence Speech recognition Signal representation / {4}
Natural language – processing Speech recognition Verbal interference / {5}
{All keywords} / {}

Figure 2. Lattice £(D′, K′, I′) of the formal context C′ from the Figure 1.

At the second stage, we elicit the relevant keywords which are associated with the newly added document *d*. Then, a weight for each relevant keyword is calculated by definition 5. Then, the ordered relevant keywords are presented to the user with their relevant weight. The user can select any that are relevant. The user can also view the sub-lattice and the relevant documents for each of the relevant keywords during this process. The similarity relation between keywords and documents can be easily observed through the lattice.

For example, let a new document *d* be '5' and the set of keywords (Γ) of *d* be {natural language processing, speech recognition, verbal interference}. Then, we can get a set of documents associated with Γ (Δ_Γ) ={3, 4} by definition 3 from the sub-context C′ = (D′, K′, I′) shown in figure 2. After that, the set of keywords which are associated with Δ_Γ is obtained: that of $f(\Delta_\Gamma)$ = {artificial intelligence, information retrieval, natural language processing, speech recognition, signal representation} by definition 4. Finally we define the set of relevant keywords as $\Re = f(\Delta_\Gamma) - \Gamma$ = {artificial intelligence, information retrieval, signal representation}. Because the set of keywords in $\Re$ are candidates for expanding the keywords already associated with *d*. Then, for each element of $\Re$, a frequency is calculated by definition 5 as follows: *Freq*(Γ, artificial intelligence)=3, *Freq*(Γ, information retrieval)=1 and *Freq*(Γ, signal representation)=1. Through this process, the user can capture some relevant keywords (here, the keyword 'artificial intelligence' or may others) in adding a new document.

When the above stage is complete the document is located at a node. If there is another document(s) already at the node, the user adding the new document is presented with the previous document and asked to include keywords that distinguish the documents. The user can chose to leave the two documents together with the same keywords. Ultimately however, every document is unique and offers different resources to other documents and probably should be annotated to indicate the differences. The approach used here is derived from Ripple-Down Rules, but the location of the document is determined by the lattice structure rather than the history of the development.

In the RDR approach, when a new rule is added, all stored cases that can be reach the parent rule (cornerstone cases) are retrieved. Then the user is required to construct a rule which distinguishes between the new case and the cornerstone cases until it excludes all cornerstone cases. In our document retrieval system, a case which has the same set of keywords as the new document, becomes the equivalent of a cornerstone case. If a cornerstone case exists, the system displays all the keywords used by other annotators. The user should select at least one different feature (keyword) from the deployed keywords or specify a new word to distinguish the cornerstone case(s) from the new case. Added keywords will result in moving to a lower node, where other cases may be found those have other parent nodes. This process is continued until the user is satisfied.

Another knowledge acquisition issue we have addressed is when a new term is entered for a new document; this term may also appropriately apply to other documents already in the system. This problem could be left until the system fails to provide an appropriate document for a later search as in the RDR approach. However, in our approach, the system passes a log of the addition of a new document to a meta-expert. The expert can then decide whether any document at the parent nodes for the new nodes should also have the term added. The following definitions are used in identifying the relevant documents and their associated terms.

Definition 6: Let £ = < V, ≤ > be a lattice. Given a node $\theta \in V$, the set of direct parents of θ denoted $DP_£(\theta)$ is defined as follows: $DP_£(\theta) = \{\alpha \in V \mid \theta < \alpha$ and there does not exist any $\beta \in V$ such that $\theta < \beta \;\&\; \beta < \alpha\}$.

Definition 7: Let £(C) be a concept lattice of the formal context C = (D, K, I) and *d* be the new document. For each document $\delta \in D$, we can define the set of relevant keywords for δ with respect to *d* denoted $Rel_d(\delta)$ as follows:

$$Rel_d(\delta) = \Big\{ f(d) \setminus \bigcup_{(X,Y)\in DP_{£(C)}(<\{d\},\, f(d)>)\;\&\;\delta\in X} Y \Big\}$$

For example, suppose that a new document *5* (*d*) with the set of keywords Γ {natural language processing, speech recognition, verbal interference} is added into the lattice shown in Figure 1. Then the lattice structure will be reformulated to cope with the new case. Figure 2 can be a part of a reconstructed lattice which has a new added node ζ ({*d*},*f*(*d*)), coloured grey. Now, for the documents located in the direct parent node of ζ (here the document *4*), we extract the relevant keywords with respect to *d* by definition 7: Rel_5 (*4*) = {verbal interference}. The system then passes this case to the expert to be able to examine whether document *4* should have the keyword *'verbal interference'*. The reason for dealing only with the direct parent nodes is that the parents of the direct parent nodes had been observed when the direct parent nodes were added.

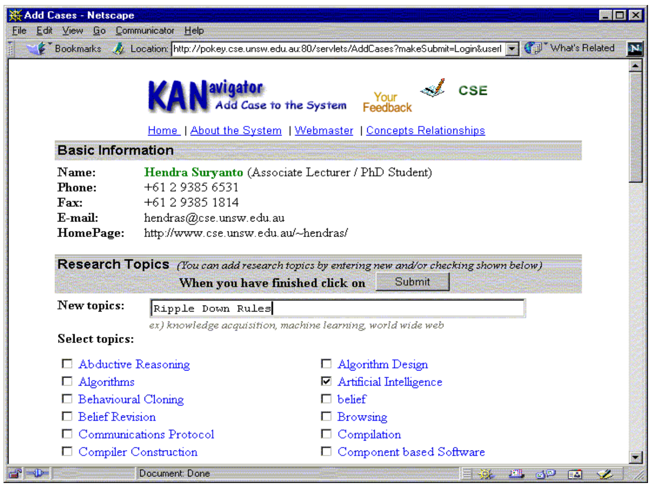
As the system evolves, new terms are being added. As a consequence, there is a need to handle synonyms or to group relevant terms together to facilitate the user's query. For this reason, we support a tool for experts to build a thesaurus for the domain as required.

Another mechanism is used when the system can not find an appropriate node in the lattice with a query. In this case, the system sends a log file to an expert so s/he can decide if more appropriate keywords are required for documents. If necessary the expert sends e-mail to the author (annotator of the document) giving the hyperlink to the node with the documents that may need changes. All interactions between the system and users are also logged for future of evaluation.

IMPLEMENTATION

A prototype has been implemented (http://pokey.cse.unsw.edu.au/servlets/Search) and demonstrated with a test domain of around 200 papers from the Banff Knowledge Acquisition Workshops. Another test domain (http://pokey.cse.unsw.edu.au/servlets/RI) is for research topics in the School of Computer Science and Engineering, UNSW. There are around 150 research staff and students in the School who generally have homepages indicating their research projects. The aim here was to allow staff and students to freely annotate their pages so that they would be found appropriately within the evolving lattice of research topics. The goal is a system to assist prospective students and potential collaborators in finding research relevant to their interests.

As already mentioned, a browsing scheme of our approach and its implementation has been introduced in the papers [16, 17]. The focus here is the knowledge acquisition mechanisms for incremental development as browsing evolves. The mechanisms described in the previous section have been implemented and are described here with reference to the domain of research topics and researchers' homepages. In this domain, a document corresponds to a homepage and a set of keywords is a set of research topics. A researcher can annotate his/her own homepage with a set of research topics by selecting among the displayed topics


Add Cases - Netscape
File Edit View Go Communicator Help
Bookmarks Location: http://pokey.cse.unsw.edu.au:80/servlets/AddCases?makeSubmit=Login&userl
What's Related
KANavigator Add Case to the System
Your Feedback
CSE
Home | About the System | Webmaster | Concepts Relationships
Document: Done

Basic Information

Name: Hendra Suryanto (Associate Lecturer / PhD Student)
Phone: +61 2 9385 6531
Fax: +61 2 9385 1814
E-mail: hendras@cse.unsw.edu.au
HomePage: http://www.cse.unsw.edu.au/~hendras/

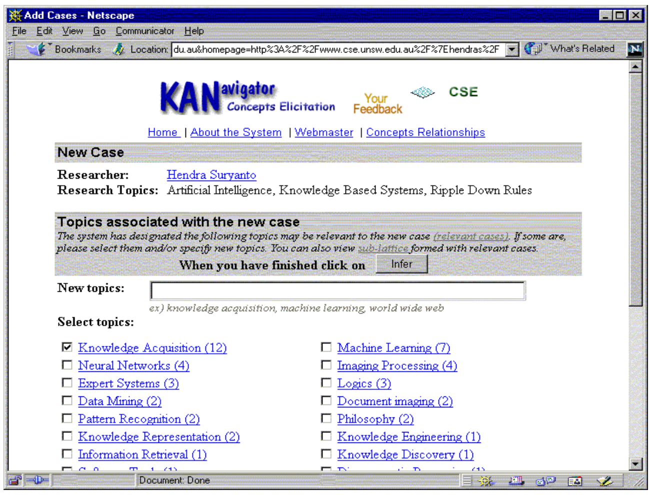
Research Topics *(You can add research topics by entering new and/or checking shown below)*
When you have finished click on Submit

New topics: Ripple Down Rules
ex) knowledge acquisition, machine learning, world wide web

Select topics:

☐ Abductive Reasoning	☐ Algorithm Design
☐ Algorithms	☑ Artificial Intelligence
☐ Behavioural Cloning	☐ belief
☐ Belief Revision	☐ Browsing
☐ Communications Protocol	☐ Compilation
☐ Compiler Construction	☐ Component based Software

Figure 3. A partial screen of annotating a homepage.


Add Cases - Netscape
File Edit View Go Communicator Help
Bookmarks Location: du.au&homepage=http%3A%2F%2Fwww.cse.unsw.edu.au%2F%7Ehendras%2F
What's Related
KANavigator Concepts Elicitation
Your Feedback
CSE
Home | About the System | Webmaster | Concepts Relationships
Document: Done

New Case

Researcher: Hendra Suryanto
Research Topics: Artificial Intelligence, Knowledge Based Systems, Ripple Down Rules

Topics associated with the new case
The system has designated the following topics may be relevant to the new case (relevant cases). If some are, please select them and/or specify new topics. You can also view sub-lattice formed with relevant cases.
When you have finished click on Infer

New topics:
ex) knowledge acquisition, machine learning, world wide web

Select topics:

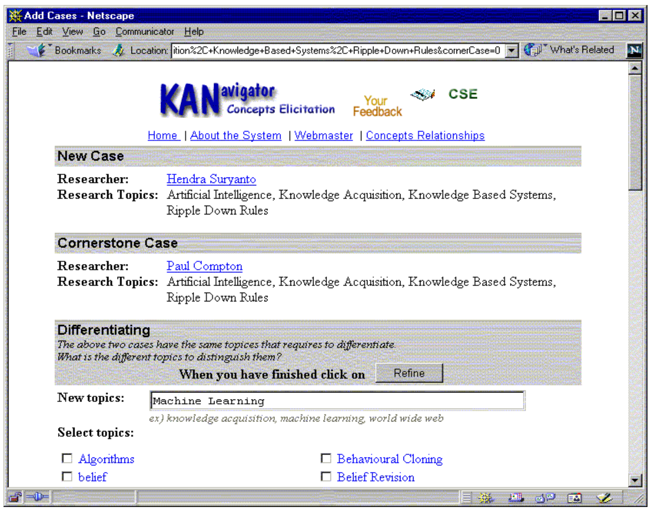
☑ Knowledge Acquisition (12)	☐ Machine Learning (7)
☐ Neural Networks (4)	☐ Imaging Processing (4)
☐ Expert Systems (3)	☐ Logics (3)
☐ Data Mining (2)	☐ Document imaging (2)
☐ Pattern Recognition (2)	☐ Philosophy (2)
☐ Knowledge Representation (2)	☐ Knowledge Engineering (1)
☐ Information Retrieval (1)	☐ Knowledge Discovery (1)

Figure 4. A partial screen displaying possibly relevant topics for the case being annotated.

or by specifying new topics through the interface shown in Figure 3. The system displays all the topics used by other researchers.

Topics are initially selected by clicking the checkbox, or entering a new topic. Then, the weight for related topics is calculated and a shorter list of these related topics is presented shown in Figure 4. Again the user simply clicks the check box located in the front of each topic. At this stage the user can view the ordered set of documents and a sub-lattice constructed of relevant documents by clicking a hyperlink on this screen. The user can also view the documents for each of the related topics as well as the existing lattice structure. Through these processes, the user may find other relevant topics s/he has missed from the initial long list presented.

When the above stage is complete the document (homepage) is located at a node in a lattice. If there is another document(s) already at the node, the user is presented with the previous document(s) and given the opportunity to include topics that distinguish the documents. The user should select at least one different feature (topic) to distinguish the previous document(s) from


Add Cases - Netscape
File Edit View Go Communicator Help
Bookmarks Location: ition%2C+Knowledge+Based+Systems%2C+Ripple+Down+Rules&cornerCase=0
What's Related
KANavigator Concepts Elicitation
Your Feedback
CSE
Home | About the System | Webmaster | Concepts Relationships

New Case

Researcher: Hendra Suryanto
Research Topics: Artificial Intelligence, Knowledge Acquisition, Knowledge Based Systems, Ripple Down Rules

Cornerstone Case

Researcher: Paul Compton
Research Topics: Artificial Intelligence, Knowledge Acquisition, Knowledge Based Systems, Ripple Down Rules

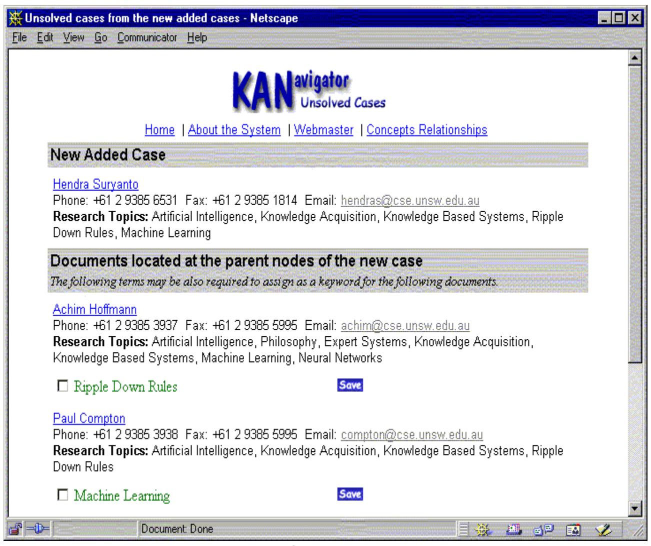
Differentiating
The above two cases have the same topices that requires to differentiate.
What is the different topics to distinguish them?
When you have finished click on Refine

New topics: Machine Learning
ex) knowledge acquisition, machine learning, world wide web

Select topics:

- ☐ Algorithms
- ☐ Behavioural Cloning
- ☐ belief
- ☐ Belief Revision

Figure 5. An example of differentiating two cases.


Unsolved cases from the new added cases - Netscape
File Edit View Go Communicator Help
KANavigator Unsolved Cases
Home | About the System | Webmaster | Concepts Relationships
Document: Done

New Added Case

Hendra Suryanto
Phone: +61 2 9385 6531 Fax: +61 2 9385 1814 Email: hendras@cse.unsw.edu.au
Research Topics: Artificial Intelligence, Knowledge Acquisition, Knowledge Based Systems, Ripple Down Rules, Machine Learning

Documents located at the parent nodes of the new case
The following terms may be also required to assign as a keyword for the following documents.

Achim Hoffmann
Phone: +61 2 9385 3937 Fax: +61 2 9385 5995 Email: achim@cse.unsw.edu.au
Research Topics: Artificial Intelligence, Philosophy, Expert Systems, Knowledge Acquisition, Knowledge Based Systems, Machine Learning, Neural Networks

☐ Ripple Down Rules Save

Paul Compton
Phone: +61 2 9385 3938 Fax: +61 2 9385 5995 Email: compton@cse.unsw.edu.au
Research Topics: Artificial Intelligence, Knowledge Acquisition, Knowledge Based Systems, Ripple Down Rules

☐ Machine Learning Save

Figure 6. An example of cases and topics related to the new case.

the new case. A further document may be prompted by the new added topics and the process can be continued. Figure 5 shows an example of this stage.

Next, when a new term is entered for a new document; this term may also appropriately apply to other documents already in this system. The expert considers whether any document at the parent nodes for the new node should also have the term added. However, may be too costly to go through every case where the terms of the new case apply. Figure 6 shows an example of this stage.

Another mechanism is motivated from when the system can not find a node in the lattice with a query. In this case, the system sends a log file to an expert so s/he can decide if more appropriate keywords are required for the documents. If the expert makes a decision for the case(s), the system creates e-mail automatically and sends it to the author (annotator of the document) by attaching a hyperlink which can facilitate the refinement of the keywords of the document if desired.

We have not presented an example of keeping a log file referring this to the expert as this is straight forward. Nor have we included an ontology example as this is beyond.

An end-user searching for a research area of researcher can specify a query by entering any textwords in a conventional information retrieval fashion or by selecting a term among those already used for annotating documents. A set of words can be entered separated by commas (′,′) assuming the AND Boolean operator. When a query is entered, stopwords are first eliminated and the remaining query stemmed using the stemming classes. Next, the system tries to identify the most relevant portion in the lattice for the query and moves to this node displaying only the direct neighbours of the node. The user can start navigation from the node. If the system cannot find a node in the lattice with the query, documents (researchers' homepages) will be retrieved which contain these textwords and system formulates a sub-lattice using the results and their research topics. Navigation can be done on this sub-lattice, or the user can move to a view of the whole lattice. In this situation, the system sends a log file to an expert so s/he can decide if more appropriate research topics should be included for the documents.

The user can also narrow the results using the multi-valued attributes. We build a concept lattice using the result documents (homepages) with their topics 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 rather than a line diagram. 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. The user can also group Boolean attributes in the one-valued contexts (here a set of topics) by building ad-hoc taxonomies. The nested attributes associated with the user's query are built from these taxonomies. Once again the detailed implementation of these browsing mechanisms and interfaces can be found in the papers [16, 17].

DISCUSSION

Having completed a prototype implementation of the approach, it seems clear that it facilitates browsing and that users adding documents enjoy seeing how their document fits into the lattice and are motivated to make sure it is appropriately positioned. A more substantial evaluation is being undertaken based on logging the activity of both users searching for research topics and researchers changing the annotation of their home pages.

FCA is widely used for knowledge acquisition to discover concepts and rules related to objects and their attributes. Its advantage comes from the way it shows how the presence or absence of attributes distinguishes objects in the various super-concept sub-concept relations. In common with most KA techniques, its power is in the way it presents relationships across the whole domain, and as noted above most FCA work attempts to display the whole lattice. This contrasts with Ripple-Down Rules (RDR), which is based on the idea that experts should only be required to distinguish between cases (or documents) in a specific context [7]. Both FCA and RDR are related to the Repertory Grid approach [10] where the expert is asked to gradually build up axes of differentiation (constructs) between objects. This is motivated by considering specific objects or cases, but as the knowledge base develops a view of the overall relationships becomes more important.

The key extension to FCA that we have implemented is similar to both RDR and Repertory grids. The system retrieves other existing documents that are annotated with the same terms as the new document being added and suggests that the expert may wish to distinguish these documents. The second key extension to FCA is that when documents are added, other keywords are suggested that are co-occur with the keywords that user has selected, elsewhere in the system.

The further major extension we propose is in the area of ontologies: either constructed by an expert for this particular system, or an ontology imported from elsewhere. First consider a simple taxonomy. The only browsing mechanism we propose is FCA so there is little point in considering inheritance mechanisms as part of a reasoning mechanism. Rather inheritance is a question for knowledge acquisition. When a new document is added and the user enters a term that occurs in the taxonomy all parents of the term up the hierarchy are also displayed. Any of these terms can be selected by the user and added to the document as terms. That is, the various superclasses are considered as Boolean attributes. The user is free to select none, one or some of these terms to add to the document. There may be multiple taxonomies available. The user is free to select any combination of superclass terms to be added to the case.

There are good reasons for having taxonomies, but the initial motivation for developing mechanisms for combining both inheritance and heuristic reasoning was probably to avoid a user having to enter the entire taxonomy. E.g. if the user had already entered “dog” it was inappropriate to ask them to enter “mammal’ and “animal’ as well to allow the appropriate rules could fire. Here however we are not asking the user browsing the system to enter such terms, but the user or expert who is entering a document. Secondly, the hierarchy is used to suggest which terms may be relevant.

We further anticipate that more complex ontologies can be handled similarly. A simple example illustrates the point. The $(KA)^2$ ontology has the attribute “author” which may have one or more values. The Boolean attribute “has_author” can be added as a term, as well as the names of individual authors. FCA will handle this perfectly adequately. This also means that missing values will be handled unambiguously. It should be noted that we are not at all suggesting that rich ontological structures and reasoning mechanisms are not of great value. However, in the context of constructing browsing systems for an individual’s documents or documents in a specialized domain or relating to a small community, the value of an ontology is in suggesting to the expert annotating a document what terms might be suitable. We recognise that there will be significant issues in importing ontologies, as the expert will need to be presented with a simple point and click selection of terms or a series of text boxes to be (optionally) filled. The expert entering documents will not want to deal with understanding notions of attributes, values, constraints, cardinality etc and how these are all reduced to Boolean attributes.

Our final proposal is the converse of the above. We propose that the user should be allowed to group Boolean attributes and also to build a hierarchy of groups of groupings. The groupings would be named. This would allow a hierarchical representation of an 'ontology' similarly to how a browsing ontology is presented in $(KA)^2$. (At present we are using nested pull-down menus). These 'ontologies' would be constructed on the fly, but stored for future use if required. A user would be free to select any one of these ontologies to interact with the system, and use this interaction to move to a particular sub-lattice. In this area it may be possible to use some form of machine learning to select likely candidate nodes from grouping together, but we have not investigated how this may be done at this stage.

CONCLUSION

We have presented an overview of how Formal Concept Analysis may be used for document retrieval and outlined the system we have implemented. The main differences between the system we have implemented and previous work in this area is an emphasis on incremental development and evolution and tools to support this. The Web implementation we have used also provides a fairly natural environment for document management. From our experience so far with this development it is clear to us that Formal Concept Analysis is a useful way of supporting the flexible open management of documents required by individuals, small communities or in specialised domains.

We have also outlined how other ontologies might be used with the system. We see existing ontologies as useful resources to assist the user in adding documents, but that the user should not be constrained and should be able to pick and choose selections from multiple ontologies. We also see that ontologies to assist a user in accessing the system should be ad-hoc, able to be constructed on the fly but also that the user should be allowed to select from past ontologies that users have constructed.

ACKNOWLEDGMENTS

The authors would like to thank Bao Vo and Dr. Rex B.H. Kwok for helping in formalising of mathematical formulas used in definitions. This research is supported by an Australian Research Council (ARC) grant.

REFERENCES

1. Aussenac-Gilles, N., Biebow B., Szulman S. Revisiting Ontology Design: A Methodology Based on Corpus Analysis, *12th European Conference on Knowledge Acquisition and Knowledge Management* (*EKAW 2000),* Springer, 172-188, 2000.
2. Benjamins, V. R., Fensel, D., Decker, S. and Perez, A. G. (KA)²: building ontologies for the Internet: a mid-term report. *International journal of human computer studies*, Vol. 51, No. 3, 687-712, 1999.
3. Carpineto, C. and Romano, G. ULYSSES: A Lattice-based Multiple Interaction Strategy Retrieval Interface, In Blumenthal et al., *Human-Computer Interaction*, Springer Verlag, 1995.
4. Carpineto, C. and Romano, G. A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval. *Machine Learning*, 24(2), 95-122, 1996.
5. Cole, R. and Eklund, P. Application of Formal Concept Analysis to Information Retrieval using a Hierarchically Structured Thesaurus. *International Conference on Conceptual Graphs, ICCS '96*, University of New South Wales, Sydney, 1-12, 1996.
6. Cole, R. and Stumme, G. CEM - A Conceptual Email Manager, *Proceedings of the 8th International Conference on Conceptual Structure (ICCS 2000),* Darmstadt, Springer, 438-452, 2000.
7. Compton, P. and Jansen, R. A Philosophical Basis for Knowledge Acquisition. Knowledge Acquisition 2:242-257, 1990.
8. Eklund, P., Groh, B., Stumme, G. and Wille, R. A Contextual-Logic Extension of TOSCANA, *Proceedings of the 8th International Conference on Conceptual Structure (ICCS 2000),* Darmstadt, Springer, 453-467, 2000.
9. Furnas, G. W., Landauer, T. K., Gomez, L. M. and Dumais, S. T. Statistical semantics: analysis of the potential performance of key-word information systems, *Bell System Technical Journal*, 62, 1753-1806, 1983.
10. Gaines, B. and Shaw, M. Cognitive and Logical Foundation of Knowledge Acquisition. *The 5th Knowledge Acquisition for Knowledge Based Systems Workshop*, Banff, 9.1-9.25, 1990.
11. Ganter, B. and Wille, R. Conceptual Scaling, In: F. Roberts (ed.): *Application of Combinatorics and Graph Theory to the Biological and Social Sciences*, Springer, 139-167, 1989.
12. Ganter, B. and Wille, R. Formal Concept Analysis: mathematical foundations. Springer, Heidelberg, 1999.
13. Godin, R., Missaoui, R. and April, A. Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *International Journal of Man-Machine Studies*, 38, 747-767, 1993.
14. Kang, B. H., Yoshida, K., Motoda, H. and Compton, P. Help Desk System with Intelligent Interface, *Applied Artificial Intelligence*, 11: 611-631, 1997.
15. Kim, M., Compton, P. and Kang, B. H. Incremental Development of a Web Based Help Desk System, *Proceedings of the 4th Australian Knowledge Acquisition Workshop (AKAW99),* University of NSW, Sydney, 13-29, 1999.
16. Kim M. and Compton, P. Developing a domain-specific Information Retrieval Mechanism. Proceedings of the 6th Pacific Knowledge Acquisition Workshop (PKAW 2000), Eds. P. Compton; A. Hoffmann; H. Motoda; T.Yamaguchi, Sydney Australia, 189-206, 2000.
17. Kim M. and Compton, P. A Web-based Browsing Mechanism Based on the Conceptual Structures, Conceptual Structures: Extracting and Representing Semantics, *Proceedings of the 9th International Conference on Conceptual Structures (ICCS'01),* Stanford University, California, USA, 47-60, 2001.
18. Maedche A., Staab S. Mining Ontologies from Text, *12th European Conference on Knowledge Acquisition and Knowledge Management* (*EKAW 2000),* Springer, 189-202, 2000.
19. Marchionini, G. and Shneiderman, B. Finding facts vs. browsing knowledge in hypertext systems, *IEEE Computer,* 21, 70-80, 1988.
20. Priss, U. 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, 2000.
21. Richards, D. and Compton, P. Knowledge acquisition first, modelling later, *Knowledge Acquisition, Modeling and Management*, E. Plaza and R. Benjamins, Berlin, Springer: 237-252, 1997.
22. Rock, T. and Wille, R. Ein TOSCANA-System zur Literatursuche, In: G. Stumme and R. Wille (Hrsg.): Begriffliche Wissensverarbeitung: Methoden und Anwendungen, Springer, Berlin-Heidelberg, 239-253, 1999.
23. Stumme, G. Hierarchies of Conceptual Scales. *12th Banff Knowledge Acquisition, Modelling and Management,* Eds. B Gaines; R Kremer; M Musen, Banff Canada, 16-21 Oct., SRDG Publication, University of Calgary, 1999.
24. Wille, R. Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ivan Rival (ed.), Ordered sets, Reidel, *Dordrecht-Boston*, 445-470, 1982.
25. Wille, R. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23, 493-515, 1992.