Incremental Development of Browsing for 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
Browsing is being supported in many information retrieval systems to supplement Boolean querying. We have implemented a web-based browsing mechanism for a domain-specific document retrieval system based on the concept lattice of Formal Concept Analysis. In this paper, we have proposed and implemented an incremental development of browsing by combing Formal Concept Analysis (FCA) and Ripple Down Rules (RDR) as well as incorporating a domain ontology. It allows a user to formulate browsing in a more systematic and general way for the domain by discovering relevant concepts when a new document is added and an existing document is refined. The user can fairly easily add new documents and annotate these so that they can be readily retrieved and also distinguished from less relevant documents. The mechanism incorporates an ontology for the documents which emerges over time rather than having to be defined from the outset. A part of experimental evaluation of the system shows good retrieval performance and a significant improvement after the introduction of an incremental knowledge acquisition mechanism.

Keywords
Information retrieval system, document retrieval system, browsing, incremental knowledge acquisition, formal concept analysis, ripple down rules

INTRODUCTION
In information retrieval processes, the degree of user interaction is typically twofold. In general, a user sends a Boolean query to the system and the system returns a result. Then the user can refine or reformulate the query corresponding to the result until he/she is satisfied with the result. In a second style of interaction, the user simply navigates a hierarchical classification scheme or subject categories displayed by the system using hyperlinks menus or some direct representation of the hierarchy.

The method of direct query formulation is most useful when the user knows what she/he is looking for and has some background knowledge of the search domain. However, setting up an appropriate query is a difficult process, especially for novice users and when the users do not know exactly what they want and how to get it.

The browsing approach is being supported in many information retrieval systems to resolve this problem. The obvious advantage of this method is that the users can quickly explore the search domains and can easily acquire the domain knowledge (Marchionini and Shneiderman 1988). Typically, browsing is formulated in a hierarchy using some sort of clustering algorithm. Thus, the effectiveness of browsing depends deeply on how well the used algorithm groups the relevant documents into the same cluster. Hierarchical Agglomerative Clustering (HAC) algorithms are probably the most commonly used clustering algorithms in information retrieval. However, this paradigm can cause the problem of category mismatch (Furnas et al. 1983; Godin et al 1993) where a wrong decision can be critical in failing to find the right documents and contributes to the low performance of the approach. This is because the clustering only formulates relationships between parent and child cluster, but not other relationships between clusters in the different branches of the hierarchy.

To solve the problem a new browsing mechanism has been introduced based on the concept lattice of the Formal Concept Analysis (FCA) (Godin et al. 1993; Carpineto and Romano 1996; Priss 1997, 2000; Kim and Compton 2001). Godin et al (1993) and, Carpineto and Romano (1996) have addressed the advantage of the lattice method against the hierarchical classification. The significant advantage of this approach is that the mathematical formulas of FCA can construct the conceptual structure which has generalisation and specialisation relationships among the concept nodes. This lattice structure allows one to reach a concept node 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.

The effectiveness of browsing also depends on whether the relationships between the clusters are constructed in a systematic and general way. Hence, most of the clustering incorporates the use of subject categories or thesauri, rather

than only using the document information itself. It is quite natural that the hierarchy can be structured more systematically when domain terms are involved.

For the same reason, information retrieval based on FCA often incorporates the use of thesauri or classification for the domains (Carpineto and Romano 1996; Cole and Eklund 1996; Stumme 1999; Priss 2000; Kim and Compton), even though the formulas of FCA construct generalised and specialised relationships in the lattice. 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. Others (Cole and Eklund 1996; Stumme 1999; Priss 2000) also use a domain thesaurus for their retrieval processes.

Following this paradigm, we have been implemented a web-based browsing mechanism for a domain-specific document retrieval system (Kim and Compton 2000, 2001). We have demonstrated the system with a test domain (URL: http:// pokey.cse.unsw.edu.au/servlets/Search). The key difference in our approach was that we focused on developing a web-based user interface which can be very natural for Web users. Thus, 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.

However, through our experiments and implementation, we have identified three key requirements for domain-specific information retrieval systems.

1. **Incremental knowledge acquisition**

It is 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 documents can be added in a batch, but it is more likely that documents will be added individually. Thus, Godin et al (1995), Carpineto and Romano (1996), and Kim and Compton (2001) proposed incremental algorithms for updating the concept lattice. All these approaches simply reconstruct the lattice incrementally to cover the new case with the given keyword set.

However, none of studies has been done in regard to incremental knowledge acquisition. When an expert assigns the set of keywords for a document, some concepts may not be made up in just the context of the input case but also can be prompted in regard to the stored cases. The expert also tends to ignore the most general concepts (keywords) of the domain, even if they (or authors) are the most appropriate agents to assign meaningful concepts for the documents. Thus, it is necessary to have certain knowledge acquisition mechanisms to be able to extract the concepts which are missed or unknown when concepts are assigned, enabling experts to constantly improve the system's retrieval.

Thus, we have proposed and developed an incremental knowledge acquisition mechanism to extract some important concepts for the cases as well as discover the domain ontology from the situated cognition view. It has been facilitated by combining the FCA browsing and RDR knowledge acquisition techniques.

2. **Discovery of a domain ontology**

Here, an ontology can be a thesaurus or a taxonomical ontology for the domain[1]. Typically, thesauri are used for indexing (browsing) to select the most appropriate thesaural entries for representing the document. They are also used for background knowledge to expand the users' query to enhance the retrieval process. In general, ontologies for the domain are established prior to developing information retrieval systems.

In this approach, we can anticipate that there will be the typical problems of software engineering methodologies in building and maintaining the ontologies, even though the approach has certain benefits. Moreover, new cases (documents) are added into the system in continually and the domain knowledge is in a dynamically changing environment. Hence, in our view the ontologies should be discovered from the domain knowledge as documents are added, or when documents fail to be retrieved, by the domain experts adding concepts rather than simply using a predefined ontology. Of course, in the pre-defined approach, the ontology can be refined, but the maintenance of ontology is still ongoing research (Benjamins et al. 1999). An interesting fact is that none of the thesaurus or classification schemes is the same for the same domain. From this fact we can anticipate that the situated cognition view of knowledge acquisition also applies to building an ontology, with more emphasis on the significance of context.

The incremental concept formation called learning from observation is also a fundamental process of human learning since the concepts to learn are not pre determined by an expert and the instances are not pre-classified with respect to these concepts.

In the involvement of thesauri or ontologies, a more critical situation is as follows: In building an ontology, the relationship between terms are defined with an equivalent (synonym) relationship, part-of and part-whole (non-hierarchical) relationship, is-a (hierarchical) relationship and so on. Here, we will only look at this problem in the hierarchy. Let C = (D, K, I) be a formal context as in Definition 1 in the following section. We start from the subsumption hierarchy of an ontology with a partially ordered set (K, $\leq$) and a context (D, K, I). D is a set of documents which are collected from the set D and K is the

[1] We use the term 'ontology', 'taxonomical ontology', and 'thesaurus' interchangeably, even these have a slightly different definition. But we prefer to use the term 'ontology', even though there is a tendency to have a meaning of 'taxonomical ontology' and 'thesaurus'.

set of terms in the ontology. When a thesaurus is involved in an information retrieval process, the following compatibility condition is assumed for a subsumption hierarchy (Carpineto and Romano 1996; Cole and Eklund 1996; Stumme 1999)

$\forall d \in D,\ \ k, i \in K:\ \ (d, k) \in I,\ k \leq i \Rightarrow (d, i) \in I$ [2]

However, the compatibility condition is not always transitive or inheritable for the instances in ontologies, even the terms themselves (k, $i \in K$) have is-a relations in the hierarchy. For example, we suppose there is a document $d \in D$ and two terms (Web servers, Java) $\in K$ with the relationship of 'Java < Web servers' from Figure 4 and suppose the document d is associated with the term 'Java" ((d, Java) $\in I$). But the document d may be in connection with the term 'Web servers' or may not ((d, Web servers) $\in I$) or ((d, Web servers) $\notin I$)).

Again, the ontologies should be discovered from the domain knowledge in connection with adding a new case and refining stored cases. We propose an approach to solve this problem by combining the knowledge acquisition process for the documents. We also provide a tool for experts to develop a domain ontology to cover the stored documents of the knowledge base by re-using keywords if required.

3. **More generalised browsing structure**

We already mentioned that effectiveness of browsing depends closely on how well the relationships between the clusters are constructed in the hierarchy in a more general and specific way. Hence, most of the clustering incorporates the use of classification schemes or thesauri. We have observed that our knowledge acquisition mechanism allows the construction of a more generalised and structured browsing scheme improving its retrieval performance. Such a browsing structure can also represent the domain ontology in the longer term.

We have implemented a browsing mechanism and an incremental knowledge acquisition mechanism for a domain-specific document retrieval system on the Web with a test domain of papers at the Banff Knowledge Acquisition Workshops in recent years (URL: http://pokey.cse.unsw.edu.au/servlets/Search). We have also observed that the system allows enabling experts to constantly improve the system's retrieval. In this paper, we focus on explaining the incremental knowledge acquisition mechanism we have developed, rather than demonstrating the system.

In the next section, we will explain methods we have used. Then we will present the incremental knowledge acquisition mechanism. Finally we will outline future directions for this work.

METHODS

One of the very complex tasks in AI is known as knowledge acquisition accompanied by the knowledge acquisition bottleneck. To improve the bottleneck problem, a new modern approach (Richard and Compton 1998; Tecuci 1998) emerged with an emphasis on the situated cognition view to incremental construction of knowledge in the context of its use, rather than transfer of knowledge. RDR uses these approaches and attempts to address incremental knowledge acquisition from a situated cognition perspective (Compton and Jansen 1990).

FCA is used for a knowledge acquisition process (Wille 1992; Erdmann 1998; Stumme 1998) 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 is still to give the expert a view of the whole domain so that all relevant concepts will be included. Despite that, 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 a 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. Thus, we tried to accomplish the knowledge acquisition process from the FCA features based on the basic philosophy of RDR. By combining the RDR and FCA techniques the expert is able to achieve incremental maintenance of the system's knowledge improving the quality of the retrieval over time.

Ripple Down Rules (RDR)

RDR is an effective knowledge acquisition and maintenance methodology which allows a domain expert to build and maintain knowledge based systems very simply and to acquire domain knowledge easily and quickly. The approach was initially developed in dealing with the problems found in the maintenance of the medical expert system GARVAN-ESI (Compton et al. 1989). The main observation in this study was that experts never gave a comprehensive explanation of why one conclusion should be given rather than another. Rather they are good at creating justifications for their decision in the context. Taking this experience to address knowledge acquisition from a situated view of the nature of knowledge, the development of RDR was started (Compton and Jansen 1990).

In the RDR method, the expert is only required to identify features that differentiate between a new case being added and the other stored cases already correctly handled. That is the main technique of knowledge acquisition in RDR which

[2] Gerd Stumme describes this problem in the paper (Stumme 1999). We explain the problem following the notion of the paper.

is very similar to the use of differences in personal construct psychology (Gains and Shaw 1990). A rule is only added to the system when a case has been given a wrong conclusion. Any cases that have prompted knowledge acquisition are stored along with the knowledge base. RDR does not allow the expert to add any rules which would result in any of these stored cases being given different conclusions from those stored. It means that the existing rules' consistency is kept in RDR and that there is incremental improvement in the system.

It has been applied to a range of tasks: multiple classification, control, knowledge reuse, heuristic search, configuration and information retrieval. There are a number of other lines of RDR research integrating RDR with machine learning and fuzzy reasoning.

Formal Concept Analysis (FCA)

FCA is a data analysis method for explicitly investigating and processing given information based on a mathematical theory (Wille 1982; Ganter and Wille 1998). It has been applied to a variety of areas for data analysis, information retrieval, knowledge acquisition and knowledge discovery in databases.

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. All formal concepts are found using given mathematic formulas. Then the subconcept-superconcept relationships between formal concepts are expressed in a concept lattice. The concept lattice can be seen as a semantic net providing "hierarchical conceptual clustering of the objects… and a representation of all implications between the attributes" (Wille 1992). More detailed definitions and examples can be found in (Ganter and Wille 1999). Here, we briefly explain it by applying it to our system.

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 our application we suppose that documents correspond to objects and the keywords of the documents constitute attribute sets. Then we define a formal context (C) as follows in our document retrieval system:

Definition 1[3]: 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, Figure 1 shows the formal context of C where D is {1, 2, 3, 4}, K is {artificial intelligence, expert systems, information retrieval, machine learning, decision tree, natural language processing, discourse analysis, speech recognition, signal representation} and the relation I is {(1, artificial intelligence), (1, information retrieval),..., (4, natural language process), (4,speech recognition), (4, signal representation)}.

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

Figure 1. Part of formal context in our application.

The following derivation is used to cultivate 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 extend and the intend of the concept (X, Y).

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

Concept Lattice

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 2.

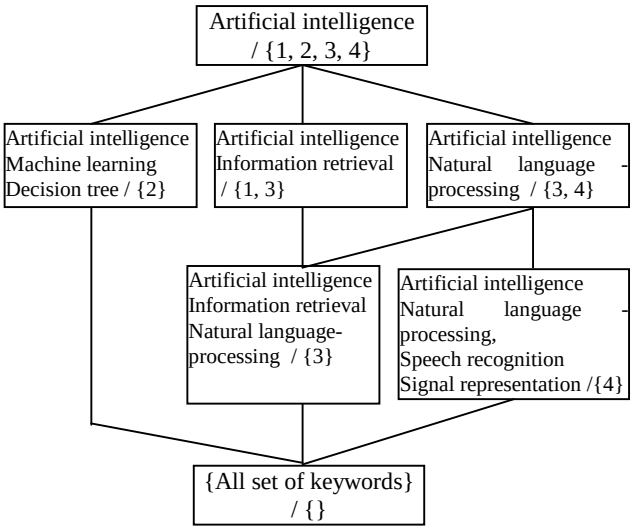

Figure 2. Concept lattice of the formal context in Figure1.

[3] This definition follows the Basic Theorem of FCA (Ganter and Wille 1999). The notion of a formal concept and concept lattice described in this paper also follow Basic Theorem of FCA.

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)

INCREMENTAL KNOWLEDGE ACQUISITION

The knowledge acquisition for the system is achieved by adding new cases and refining the existing cases. Another way of knowledge acquisition is carried out in connection with an ontology when a new case or a new class of the ontology is added. In our system, a case consists of a document, a set of keywords and other information such as authors, publication year, proceeding title and so on.

When a New Document is Added

Through the given user interface, the expert/user can add a new case and refine the stored cases. Table 1 shows the knowledge acquisition mechanism when a new document is added to the system. The formalised definitions are used in the mechanism, which is explained in detail in the following example.

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 introduced 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$\rightarrow 2^K$ 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 $f(\Delta_\Gamma) = \bigcup_{d \in \Delta_\Gamma} f(d)$ 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* (Γ, k) = $\sum_{d \in \Delta_k} |f(d) \cap \Gamma|$ where |X| is the cardinality of X.

Table 1: Knowledge acquisition: adding a document

Begin

Input a new case (document d with a set of keywords Γ)

Step1 (knowledge acquisition based on the lattice structure)

Step2 (knowledge acquisition based on the ontology)

Step3 (knowledge acquisition based on the RDR techniques)

Add the new case into the knowledge base

Reconstruct the concept lattice incrementally to cover - the new case

End

In the first step, an ordered set of documents and a set of keywords which are relevant to the new document are obtained. This step is divided into two stages. In the first stage, the ordered documents are shown to the user along with the different features between the new document and each of the set of documents. In the next stage, the frequency of each of relevant keywords is computed. Then, the ordered relevant keywords are presented to the user with their frequency.

To process this step, a sub-lattice £(D′, K′, I′) of the formal context C′ is constructed. The similarity relation between concepts can be easily observed through the lattice. 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 ($\{d\}$, Γ) with the newly added document d and its set of keywords Γ. Informally, 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 example, we suppose that there is a concept lattice as shown in Figure 2 and, a new document d (5) is added together with and its set of keywords Γ {natural language processing, speech recognition, verbal interference}. Then, we formulate the sub-context C′ = (D′, K′, I′) where D′ = Δ_Γ + $\{d\}$ = {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 3. The gray coloured box indicates the formal concept ζ. From the lattice we can get the document '4' at first. Because it exists in the direct superconcept of ζ in the lattice which indicates the most relevant of the document '5'. Next the document '3' is obtained. Finally, we get an ordered set of documents {4, 3} by the relevancy of the

document '5' in the lattice. The ordered documents are then suggested to the user along with the different features between the new document and each of the relevant documents. At this stage the user can look at the lattice structure itself using browsing mechanisms supported by the system. The result obtained by this process is equivalent to k-nearest neighbour algorithms.

At the next stage, we elicit the relevant keywords from which are associated with the newly added document *d*. Then, a frequency for each relevant keyword is calculated by definition 5. Following this, the keywords are ordered by their frequency and the system suggests the keywords to the user with their frequency. After that, the system asks the user the relevancy for each extracted keyword and the user can simply answer by clicking the check box located in the front of each keyword. 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 ($\Delta_{\Gamma)}$ ={3, 4} by definition 3 from the sub-context C′ = (D′, K′, I′) shown in figure 3. 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 of 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, experts can capture some relevant concepts (here, the keyword 'artificial intelligence' or may others) in adding a new document.

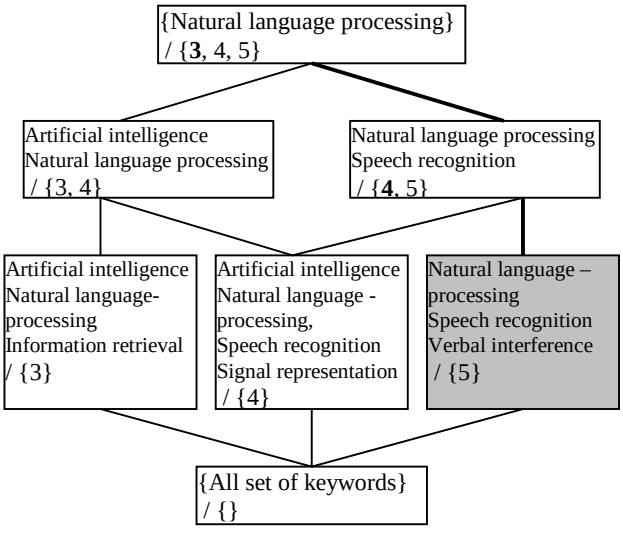

Figure 3. Lattice £(D′, K′, I′) of the formal context C′ from the Figure 2.

The next step of knowledge acquisition is based on an ontology. This mechanism will be explained in detail in the following section.

In the final stage of knowledge acquisition when a new document is added, we use the RDR techniques using a flat RDR rule tree. In the RDR approach, when a new rule is added, all stored cases that can be reached by the parent rule (=cornerstone cases) are retrieved. Then the expert 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 of the new document, becomes a cornerstone case of the new case. If a cornerstone case exists, the system elicits a relevant keyword set of the new case in the same way used in the step1 of Table 1. Then, the extracted relevant keywords become available to the expert. The expert has to select at least one different feature (keyword) from the deployed keywords or specify a new word to distinguish the cornerstone case and the new case. Another cornerstone case can be prompted by new added keywords. Thus, this process is continued until there is no cornerstone case.

When a New Class is added in the Ontology

We support a tool to develop a domain ontology to cover the stored documents of the knowledge base. Here the ontology can be a set of hierarchies of terms where a term is either a single word or a phrase along with the relationship between terms. Figure 4 shows a possible example in our document retrieval system. The structure of the ontology is also a lattice (graph), even if it presents in a hierarchy. It means the number of entries for each term exists in the structure and a term can have multi parents. For the convenience of explanation, we use the term 'class' and 'attribute' to describe a concept node in the ontology. For example, the term 'Internet' is a class with a set of attributes {Web browser, Web servers} in Figure 4. 'Web servers' is also a class with an attribute set {Apache, Java}. It means an attribute of a class can be a class of another set of attributes. As we already mentioned in the introduction, we developed a mechanism to discover new concepts when a new case is added by connecting to the process that is to be able to hold the compatibility condition in the ontological hierarchy.

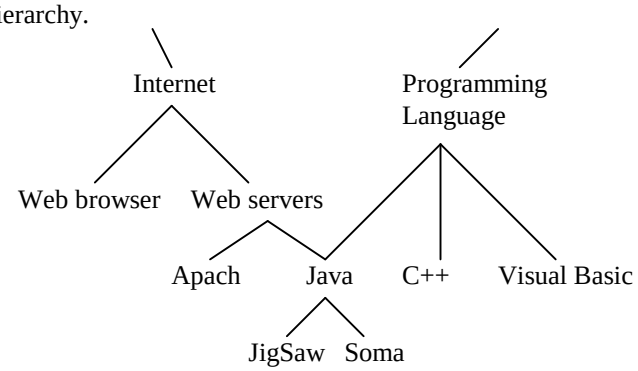

Figure 4. Part of a hierarchy of a possible ontology.

At any stage the expert/user can set up or change the hierarchy. Table 2 shows the algorithm of knowledge acquisition incorporating the ontology when a new document is added. When a document *d* with a set of keywords (Γ) is added, the system gets all class paths which the set Γ belongs to. For each class of each class-path if the document includes the class, a class is set to a value 'true' for the document, otherwise to the default value 'unknown'. For each class whose value is 'unknown' the system asks the user about inheritance between the class and the associated keyword. Then the user should answer with one of the values; 'true', 'false' or 'unknown'. After that, the system validates the assigned values of the classes for each path as follows: Let G = (V, E) be the directed graph representing the ontology and T be the set of all class paths found. Then, T is valid if and only if there does not exist any pair of nodes $u, v \in V$ such that for each path of T($v_1, \ldots, v_n$) in G the following conditions hold:

(*i*) $v_1 = u$ and $v_n = v$

(*ii*) $n > 2$

(*iii*) $v_1 =$ 'true' and $v_n =$ 'true'

(*iv*) there exists i such that $1 < i < n$ and ($v_i =$ 'false'or 'unknown')

The problem with this is not the removal of a class node in the hierarchy, but a new class (a concept node). It is probably tool costly to go through every case where the new node may apply. Thus, we just set the value of class with 'unknown' in all cases where the change in hierarchy says it might apply. Table 3 shows the algorithm of adding a new class node in ontology. Then, in any given browsing the user can choose to temporarily assign 'unknown' to either 'true' or 'false'. In other words, when the user browses the nested attributes, the system shows menu items with 'true', 'false' and 'unknown' for each attribute of the class. When the user looks at the document which belongs to the 'unknown' menu item, the system will ask the inheritance of the class in regard to the document with 'true', 'false' or 'unknown'. The knowledge base will then be changed according to the user's selection.

Table 2: Algorithm of step 2 in Table1

Begin
Get all class path (T) that the set Γ belong to
For each class path **do**
For each class in the path **do**
If the new document *d* includes the class name **then**
Set the value of the class with 'true' for *d*
Else
Set the class value with the default 'unknown';
Ask inheritance between the class and the associated keyword to the user;
(with the value of 'true', 'false' and 'unknown')
End if
End for
End for
Until validation is ok
Validate the values of the classes for each class path
If validation is ok **then**
Rebuild the set Γ by adding the classes which are set 'true';
Else
Ask the class value which is not valid again
End if
Modify the knowledge base
End

Table 3: Algorithm of adding a class in ontology

Begin
Input a class with a set of attributes
Get a set of documents that is associated with at least
- one of the attributes;
For each document **do**
If the class name exist in the keyword of the document **then**
Set the value of the class ← 'true'
Else
Set the value of the class← 'unknown'
End if
End for
For {the whole set of documents}
- {the set of documents} **do**
Set the value of the class ← 'false'
End for
End

Experimental Evaluation of Retrieval

In our experiment, we evaluated how the retrieval effectiveness of browsing can shift when the compatibility condition is held in the ontological hierarchy. Other incremental knowledge acquisition factors developed still remain to be empirically evaluated.

The experiments were carried out on a collection of documents, which belongs to the domain of the knowledge acquisition area. The collection consists of 200 documents and its taxonomical ontology. Then, we reformulated the ontology into three different hierarchies which have the average number of entries for each thesaural term with 1.12, 1.23 and 1.35 respectively by adding and pruning each term's parents. Next, we constructed three different ontologies which hold the compatibility condition with the percentage of 'unknown' value 80%, 50% and 10% respectively for each reformulated ontology. It has been conducted by relevance judgements manually. We also built a set of 40 queries which are related to the ontological terms by conjunctive and disjunctive keywords. For each query, relevance decisions with documents were also given in advance. Someone may argue with the size of the documents. But we believe the size is quite enough for the test to be considered significant for our approach. Currently we are evaluating a quite large set of documents obtained from INSPEC. The documents consist of a title, an abstract and a set of keywords with an average of 6.15. Table 4 shows the results of our experiment.

		Average number of entries for each term					
		1.12		1.23		1.35	
		Recall	Precision	Recall	Precision	Recall	Precision
Case 1: Retrieval without an ontology		0.71	0.78	0.71	0.78	0.71	0.78
Case 2: Retrieval with an ontology		0.80	0.76	0.81	0.71	0.82	0.66
Case 3: Retrieval with an ontology	80%	0.73	0.80	0.74	0.80	0.73	0.79
which holds the compatibility condition	50%	0.76	0.82	0.75	0.83	0.75	0.82
(percentage of 'unknown' value)	10%	0.79	0.85	0.81	0.84	0.81	0.85

Table 4. Average values of retrieval performance in recall and precision.

Clearly, the findings indicate that the effectiveness of retrieval both in recall and precision has improved when the ontology, which holds the compatibility condition, incorporates the retrieval against Case 1. The retrieval performance is improved when the 'unknown' values of terms are revealed. That is, the quality of the retrieval is improved by incremental maintenance of the system's knowledge over time. In the comparison with Case 2 and Case 3 in Table 4, the results show Case 3 has a better performance in precision, but a lower effectiveness in recall. However, we already pointed out that the incremental maintenance of the system increases the performance in recall. The results also reveal that the retrieval performance is proportional to the average number of ontological entries for each term in Case 2. It means that the precision of retrieval is getting worse, even though the recall is improved. In other words, the bigger the average number of entries for each term becomes, the more irrelevant documents will be.

SUMMARY

In previous work (Kim and Compton 2000, 2001), we have demonstrated a browsing mechanism that enables the user to more easily explore documents appropriate to specialised domains and based on the involvement of experts in assigning concepts to documents. We observed that experts can examine the relationships of the concepts in the lattice along with the existing document to decide whether the keywords used are appropriate and they can refine concepts of the document. We see the method as applying only to fairly small sets of keywords attached to documents by experts. In this paper, we have proposed and implemented an incremental development of browsing by combining the FCA and RDR techniques as well as incorporating a domain ontology. It allows the user to formulate browsing in a more systematic and general way for the domain by discovering relevant concepts when a new document is added and an existing document is refined. The user fairly easily adds new documents and annotates these so that they can be readily retrieved and also distinguished from less relevant documents. The mechanism incorporates an ontology for the documents to emerge over time rather than having to be defined from the outset. A part of experimental evaluation of the system shows good retrieval performance and a significant improvement after the introduction of an incremental knowledge acquisition mechanism.

Although FCA and RDR seem an attractive solution to incrementally develop a browsing mechanism for a specialised domain, we have not yet fully evaluated this approach and not yet carried out knowledge acquisition in a timely manner based on the proposed mechanism. We need to evaluate this approach in routine use with reasonably large data sets. At a more fundamental level, the value of FCA for IR is based on the assumption that when you enter a keyword, and 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 associated with browsing 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.

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.

REFERENCES

1. 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.

2. Carpineto, C. and Romano, G. A Lattice Conceptual Clustering System and Its Application to Browsing Retrieval. *Machine Learning*, 24(2), 95-122, 1996.
3. 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.
4. Compton, P., Horn, K., Quinlan, J. R., Lazarus, L. and Ho, K. (1989). Maintaining an Expert System, In J. R. Quinlan (Eds.). Application of Expert Systems, London, Addition Wesley, 366-385, 1989.
5. Compton, P. and Jansen, R. A Philosophical Basis for Knowledge Acquisition. *Knowledge Acquisition* 2:241-257, 1990.
6. Erdmann, E. Formal Concept Analysis to Learn from the Sisyphus-III Material. *Eleventh Workshop on Knowledge Acquisition, Modeling and Management* (KAW'98), Banff, Alberta, Canada, 1998.
7. Furnas, G. W. Generalized fisheye views, Proceedings of the *Human Factors in Computing Systems*, North Holland, 16-23, 1986.
8. 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.
9. 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.
10. Ganter, B. and Wille, R. Formal Concept Analysis: mathematical foundations. Springer, Heidelberg, 1999.
11. 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.
12. Godin, R., Missaoui, R. and Alaoui, H. Incremental concept formulation algorithms based on Galois (concept) lattices. Computational Intelligence, 11(2), 246-267, 1995.
13. 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. Matoda; T. Yamaguchi, Sydney Australia, 189-206, 2000.
14. Kim, M. and Compton, P. A Web-based Browsing Mechanism Based on Conceptual Structure. *will be appeared in Proceedings of 9th International Conference on Conceptual Structures (ICCS'01)*, 2001.
15. Marchionini, G. and Shneiderman, B. Finding facts vs. browsing knowledge in hypertext systems, *IEEE Computer*, 21, 70-80, 1988.
16. Richards, D. and Compton, P. Taking up the Situated Cognition Challenge with Ripple Down Rules. *International Journal of Human-Computer Studies* 49:895-926, 1998.
17. Priss, U. E. 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, 1997.
18. Priss, U. Faceted Information Representation, In: Stumme, Gerd (de.), Working with Conceptual Structures. Proceedings of the 8th International Conference on Conceptual Structures, Shaker-Verlag, Aachen, 84-94, 2000.
19. Stumme, G. Distributive Concept Exploration: a knowledge acquisition tool in formal concept analysis. In: O. Herzog, A. Gunter (eds.): KI-98: Advances in Artificial Intelligence. LNAI 1504, Springer, Berlin-Heidelberg, 117-128, 1998.
20. 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.
21. Tecuci, G. Building Intelligent Agents: An Apprenticeship Multistrategy Learning Theory, Methodology, Tool and Case Studies. Sydney, Academic Press, 1998.
22. Wille, R. Restructuring lattice theory: an approach based on hierarchies of concepts. In: Ivan Rival (ed.), Ordered sets, Reidel, *Dordrecht-Boston*, 445-470, 1982.
23. Wille, R. Knowledge acquisition by methods of formal concept analysis. In: E. Diday (ed.): Data analysis, learning symbolic and numeric knowledge. Nova Science Publisher, New York, Budapest, 365-380, 1989.
24. Wille, R. Concept lattices and conceptual knowledge systems. *Computers and Mathematics with Applications*, 23, 493-515, 1992.