Using Formal Concept Analysis with an Incremental Knowledge Acquisition System for Web Document Management

Timothy J. Everts, Sung Sik Park and Byeong Ho Kang

School of Computing,
University of Tasmania
Sandy Bay, Tasmania 7001, Australia

{tjeverts, sspark, bhkang}@utas.edu.au

Abstract

It is necessary to provide a method to store Web information effectively so it can be utilised as a future knowledge resource. A commonly adopted approach is to classify the retrieved information based on its content. A technique that has been found to be suitable for this purpose is Multiple Classification Ripple-Down Rules (MCRDR). The MCRDR system constructs a classification knowledge base over time using an incremental learning process. This incremental method of acquiring classification knowledge suits the nature of Web information because it is constantly evolving and being updated. However, despite this advantage, the classification knowledge of the MCRDR system is not often utilised for browsing the classified information. This is because it does not directly organise the knowledge in a way that is suitable for browsing. As a result, often an alternate structure is utilised for browsing the information which is usually based on a user's abstract understanding of the information domain. This study investigated the feasibility of utilising the classification knowledge acquired through the use of the MCRDR system as a resource for browsing information retrieved from the WWW. A system was implemented that used the concept lattice-based browsing scheme of Formal Concept Analysis (FCA) to support the browsing of documents based on the MCRDR classification knowledge. The feasibility of utilising classification knowledge as a resource for browsing documents was evaluated statistically. This was achieved by comparing the concept lattice-based browsing approach to a standard one that utilises abstract knowledge of a domain as a resource for browsing the same documents.

Keywords: Formal Concept Analysis, Document Management, Knowledge Acquisition, Document Browsing

1 Introduction

The World Wide Web (WWW) has become the most popular information source for people today and is now the largest sharable and searchable repository of information (Park, Kim et al. 2003; Kim and Compton 2004). Originally the WWW widely utilised a passive information delivery mechanism that meant users would have to search for and then 'pull' down the information they needed. In order to overcome this limitation, a more active mechanism was required. This stemmed the research and development of software applications that could deliver the most up to date information in a timely manner. Web Monitoring Systems are an example of such software that has become popular in recent times (Liu, Pu et al. 2000; Tang, Liu et al. 2000; Boyapati, Chevrier et al. 2002; Liu, Tang et al. 2002). They check predefined target Web pages, automatically detect changes in these pages, and prompt users when these changes occur. The use of such systems appears to offer at least a partial solution to the problems of traditional information retrieval methods such as Web search engines, because the user has more control over the type and amount of information being delivered. It also ensures that the information being gathered is the latest.

However, the quantity of information being gathered can still be reasonably large. Subsequently, an effective method for storing and managing this information is also required. Document classification is one of the solutions to this problem. Traditionally, the dominant approach for classification is based on the content (text) of documents through trained classifiers using Machine Learning (ML) techniques because they achieve impressive levels of effectiveness (Sebastiani 2002). However, although classification by ML has proved to be successful in some commercial or research applications (Mladenic 1999), it is not generally appropriate for classifying information from the WWW. This is because the classification knowledge created during the training process cannot usually cater for the dynamic nature of Web documents. New information is constantly being generated or it is being updated. For this reason, efficient classification of documents retrieved from the WWW requires a technique that can operate on a continual learning process. This enables incremental knowledge acquisition that suits the dynamic nature of Web document information (Kim, Park et al. 2004).

A technique that has been found to be suitable for this purpose is the Multiple Classification Ripple-Down Rules

(MCRDR) knowledge acquisition method. Unlike machine learning methods, MCRDR constructs a classification knowledge base incrementally over time through a process of differentiation by the expert. When the case-based reasoning system of MCRDR retrieves cases that are recognised by the expert as inappropriate, the expert simply identifies the important characteristics of the present case that distinguish it from existing cases. In this way, knowledge is acquired by the system and new rules are created accordingly. When applied to Web Monitoring Systems, this technique enables the MCRDR rule set to be developed and adapted to suit the dynamic nature of Web documents (Park, Kim et al. 2003; Kim, Park et al. 2004).

Despite the appropriateness of using MCRDR to classify the documents collected by Web Monitoring Systems, the technique has one major weakness. MCRDR does not directly organise the knowledge in a way that is suitable for browsing (Kim and Compton 2004). As a result, the heuristic classification knowledge in an MCRDR knowledge base is not often utilised for browsing and searching the documents. Instead browsing and searching is facilitated through a structure based on some form of abstracted knowledge about the document domain that has been provided by the expert or user (Park, Kim et al. 2004). Therefore, it is suggested that the classification knowledge acquired through the use of MCRDR may also provide a useful resource for browsing the retrieved documents. To this extent, our research undertaken assessed the feasibility of utilising the heuristic classification knowledge of an MCRDR knowledge base as a resource for browsing documents in a specified domain. A system was developed and implemented that adopted the lattice-based browsing method of Formal Concept Analysis (Ganter, Stumme et al. 2005) as a means of providing a browsing representation based on heuristic classification knowledge. Formal Concept Analysis has been shown by Kim (Cole 2000; Kim and Compton 2001) to be quite successful for browsing documents in a specified domain. A comparative statistical analysis was performed between the use of a traditional browsing structure (based on abstract knowledge of a domain), and the concept lattice structure of FCA (based on heuristic classification knowledge). This has been done to evaluate the feasibility of utilising heuristic classification knowledge for browsing Web documents.

2 Related Work

2.1 WebMon and MCRDR

The WebMon Web Monitoring System was developed by a number of researchers at the University of Tasmania, Australia, and was built as part of the Personalised Web Information Management System detailed in Park et al. (Park, Kim et al. 2003). A Web monitoring system needs a method for archiving collected information effectively so it can be utilised in the future. For this purpose, WebMon adopts the MCRDR knowledge acquisition technique to classify and store retrieved documents appropriately.

The Multiple Classification Ripple Down Rules (MCRDR) method is derived from the Ripple Down Rules (RDR) method, a hybrid case-based and rule-based approach for knowledge acquisition and representation (Richards 2001). Knowledge acquisition (KA) in MCRDR involves the incremental addition of cases and justifications (rules) in the circumstance where a case is misclassified by the MCRDR system in the retrieval process. This incremental approach to KA is centred on the idea that the knowledge an expert provides is essentially a justification for a conclusion in a particular context (Compton and Jansen 1989; Preston et al. 1996). When the case-based reasoning (CBR) system of MCRDR retrieves a case(s) that is incorrect, the expert is required to identify the important characteristics that distinguish the incorrectly retrieved cases from the present case (Kang, Yoshida et al. 1997). It is thought that experts will select more valid knowledge if asked to deal with the differences between cases (Kang, Yoshida et al. 1997). Thus, the expert's justification provides a basis for a new rule to be created. The new rule(s) is first validated against existing rules (cornerstone cases) and then automatically appended to the knowledge base.

The MCRDR knowledge acquisition technique is used by the WebMon Web Monitoring System for determining where documents retrieved during Web monitoring should be stored for archival and sharing purposes. The structure used by the system to store the information is a storage folder structure (SFS). It is comparable to a hierarchical tree arrangement of folders, much like that used in common operating system environments such as Microsoft Windows. Depending on the choice of the user, the entire SFS can be defined up front or it can be defined incrementally as documents are collected. It is important to note that there are no predefined specifications that state the requirements for the specific folders contained in the SFS. The structure is usually devised based on the user's knowledge or understanding of the monitored document domain. It should also be noted that if the user chooses to utilise the Web portal option to share the collected information with other users, this same storage folder structure is replicated on the Web portal site. It is provided as a means for browsing and searching for the documents.

Once the SFS has been defined, newly updated Web documents retrieved during Web monitoring are classified into one or more target folders. Keywords are extracted from documents and form the conditions of rules in the MCRDR knowledge base. The rule conclusions are target folders in the SFS. This means that keywords in a newly retrieved document can be utilised in inference the MCRDR knowledge base, in order to recommend a target storage folder for the document. In the circumstance when a document is misclassified as a result of the inference process, the user simply adds knowledge to the knowledge base that enables a correct classification to be made.

As an example of the inference process for a document, Figure 1 shows how a document with the case (keywords) of [a,b,c,d,e,f,g] is recommended to storage locations within the SFS. The MCRDR KBS is drawn as an n-ary tree, with each node of the tree representing a rule which has a corresponding case. The inference process involves all rules attached to true parents being evaluated against the data. Thus the process begins by evaluating the root

rule and then moving down level by level until either a leaf node is reached or none of the child nodes evaluate to true (Dazeley and Kang 2003). Since multiple pathways of refinement can be selected, multiple conclusions can be reached. This means that the last true rule on each pathway forms the conclusion for the case. Therefore, in the case presented in Figure 1, the inference process results in the recommendation of three storage folders for the current document, namely folders F_2, F_6, and F_5.

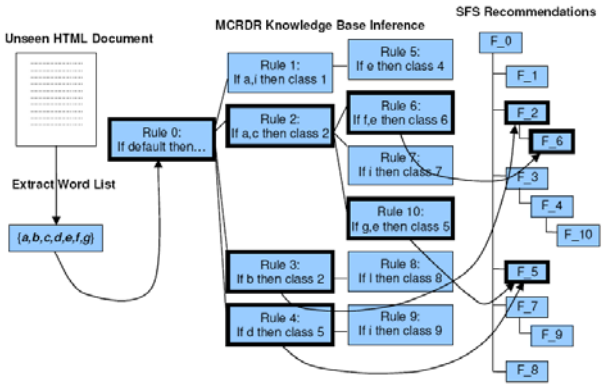

Figure 1 - Inference for a Web Document Classification

Analysis of the WebMon Web Monitoring System reveals that the user (or domain expert) is utilising the devised SFS as a basis for defining a conclusion for document classifications. The common folder structure is used as a mediating knowledge representation for the user, and it enables them to easily build a conceptual document classification model using folder manipulation. In other words, the devised SFS is an explicit representation of the user's knowledge of the current document domain. Evidently, two types of knowledge are actually being utilised in the classification process. One type of knowledge is being used to define the SFS, while another type of knowledge is being used in the actual classification of documents to target folders. This point is more apparent when the user devises the SFS. Its structure is based upon their conceptual hierarchical understanding of the domain. However, when the user classifies a document to a folder in the storage structure, that classification is made based on the actual content of the document, namely keywords. These keywords may also be embedded in the conditions of the existing classification rules in the MCRDR knowledge base. The knowledge used in the creation of the SFS is hereafter referred to as being 'abstract domain knowledge'. In regards to the second type of knowledge, it is hereafter referred to as being 'heuristic classification knowledge', since it is associated with the classification knowledge embedded in the rules of the MCRDR knowledge base. Having discovered that there are two types of knowledge being utilised by WebMon for document classification, it is well worth noting that only the abstract domain knowledge is ever utilised for browsing the documents.

Although there are two potentially useful knowledge types which could be used as a basis for browsing documents, only one of them is currently being utilised by the majority of Web portal sites. This means WWW users are being forced into searching for documents using a user-defined structure which is based on abstract domain knowledge rather than on heuristic classification knowledge. It can be argued that the heuristic classification knowledge would be more appropriate for being used as a basis for browsing the documents, because it more accurately represents the actual content of each document. For this reason, the main suggestion of this research was that if the classification knowledge can be incorporated as the basis for a document browsing structure, it may also provide an extremely useful resource for browsing the documents in the domain. Therefore, it was proposed that the use of an alternate browsing method instead of the storage folder structure may enable classification knowledge to be utilised as a basis for browsing the documents classified by MCRDR. The approach suggested and adopted in this research was the lattice-based browsing scheme of Formal Concept Analysis, so therefore it is outlined in the section that follows.

2.2 Formal Concept Analysis

Formal Concept Analysis (FCA) is a mathematical approach used for conceptual data analysis and knowledge processing. It has had numerous applications for data analysis and information retrieval in fields such as medicine, psychology, ecology, social science and political science. Various researchers have shown that a quite successful method for browsing documents in a specified domain is the lattice-based browsing approach of Formal Concept Analysis (Cole 2000; Cole, Eklund et al. 2004; Kim and Compton 2004; Becker 2005; Carpineto and Romano 2005; Eklund and Wormuth 2005; Quan, Hui et al. 2005).

FCA 'formulates concepts in terms of objects and their properties or attributes, and provides a way of combining and organising individual concepts (of a given context) into [a] hierarchically ordered conceptual structure [known as a] … concept lattice structure' (Rajapakse and Denham 2003). Correia et al. (Correia, Willie et al. 2003) comments that concepts are necessary for expressing human knowledge and a formalisation of concepts acts as means of communicatively representing knowledge.

FCA is based on a formal understanding of a concept as a unit of thought, comprising its extension and intension. The extension (extent) of a formal concept is formed by all objects to which the concept applies (a set of objects) and the intension (intent) consists of all attributes existing in those objects (a set of attributes). The set of objects, set of attributes and the relations between an object and an attribute in a data set form the basic conceptual structure of FCA (known as a formal context). A formal context is defined as a triple (G, M, I) where I maps the relation between a set of objects G, and a set of attributes M. This is denoted formally as:

$$C = (G, M, I)$$

where C represents the context. In order to express that a particular object g is in a relation I with a particular attribute m, the relation is given by:

$$(g, m) \in I \text{ or } gIm$$

and should be read as "the object g has the attribute m".

Once a formal context has been defined, all the formal concepts of the formal context can be derived. A formal concept is represented as a pair (A, B), where A is a subset of objects of the formal context and B is a subset of attributes of the formal context. In order for a pair (A, B) to be a formal concept, all attributes common to objects in A, the intent, and all objects common to attributes in B, the extent, must be the same.

This duality relationship is formalised by:

1. Set of attributes common to the objects in A (intent)

$$A' = \{ m \in M \mid (g,m) \in I \text{ for all } g \in A\}$$

2. Set of objects common to the attributes in B (extent)

$$B' = \{ g \in G \mid (g,m) \in I \text{ for all } m \in B\}$$

The formal concepts of a formal context can be ordered and arranged hierarchically into a conceptual structure of FCA called a concept lattice. Ganter and Wille (1997) comment that concept lattices are useful for unfolding given data, 'making their conceptual structure visible and accessible, in order to find patterns, regularities, exceptions etc.' Therefore, the concept lattice structure provides a means of revealing the implicit relationships between data that are not otherwise obvious. The concept lattice is ordered by the smallest set of attributes (intent) between the concepts and thus maps an ordering from the most general to the most specific concept, top to bottom (Kim 2003).

To form the concept lattice, hierarchical subconcept - superconcept relations between all the formal concepts need to be found. This is formalised by $(A_1, B_1) \leq (A2, B2) : \Leftrightarrow A1 \subseteq A2 (\Leftrightarrow B2 \subseteq B1)$ where (A1, B1) is called a subconcept of (A2, B2), and (A2, B2) is called a superconcept of (A1, B1). 'The relation $\leq$ is called the hierarchical order of the concepts' (Kim 2003, p. 55). When the lattice is formed, the largest *subconcept* will be the top most concept in the lattice, called the *supremum*, and the smallest *subconcept* will be the bottom most concept, called the *infimum*.

2.3 Combining MCRDR with FCA for Browsing Documents

Various studies have shown that the lattice-based method of FCA can be utilised as an effective means for browsing documents in specialised domains. Kim (2003) developed a Document Management and Retrieval System (DMRS) for specialised domains on the WWW that utilised an incrementally built concept lattice as a means of browsing and retrieving documents. As part of her work, a user evaluation was performed on the browsing and retrieving of documents using the lattice structure. The evaluation concluded that users considered searching a specialised domain using lattice-based browsing to be more helpful than using Boolean queries and hierarchical browsing. Furthermore, users also found that the ad hoc evolvement of the lattice-based browsing structure provided good efficiency in retrieval performance. The lattice-based browsing approach has also been shown to be much more advantageous than a hierarchical approach to browsing documents, such as the storage folder structure used by WebMon (Kim and Compton 2004). In regards to utilising MCRDR classification knowledge in the lattice structure, research undertaken by Richards (Richards 1998) revealed that the rules of an RDR knowledge base can be utilised to generate an FCA concept lattice structure. Therefore, the lattice-based browsing method of FCA may be used as a means for defining an effective document browsing structure that is based on MCRDR heuristic classification knowledge. The feasibility of this could be tested by utilising the structure to browse the documents collected by the WebMon Web Monitoring system and comparing this to browsing the same documents using the system's storage folder structure.

3 System Implantation

3.1 System Overview

In order to utilise the MCRDR heuristic classification knowledge as a basis for browsing the documents collected during the Web monitoring project, it was necessary to develop a system that implemented an alternate browsing representation. Subsequently, a system, called iWeb FCA, was developed as part of this research which utilised the MCRDR heuristic classification knowledge to generate a FCA concept lattice for browsing the documents. The iWeb FCA system generates a FCA concept lattice based on the MCRDR heuristic classification knowledge to provide an alternate browsing structure for the documents collected and classified by WebMon. In addition, the system is also capable of utilising the abstract domain knowledge embedded in the storage folder structure as a resource for generating a concept lattice. The system can be configured to generate a concept lattice using either one of the knowledge sources as a resource or it can be configured to utilise both resources at once for lattice generation.

In using the system to generate a concept lattice, it is important to note that documents are considered to constitute the objects used in FCA and the rule keywords (classification knowledge) or folder names (abstract domain knowledge) are considered to constitute the attributes. However, this approach does not strictly comply with the original formulation of FCA in which an object was implicitly assumed to have some sort of unity or identity so that the attributes applied to the whole object (e.g. a car has four wheels). As Kim (2003) states, 'clearly documents do not have the sort of unity where attributes will necessarily apply to the whole document'. However, in order to use FCA in the iWeb FCA system, the following assumptions are made. Documents correspond to objects and the rule condition keywords used to classify a document or the names of the folders in which the document is stored constitute the attribute set. A similar approach has been shown by Kim (2003) to be quite feasible.

3.2 System Functionality

3.2.1 Reducing the Amount of Documents in the Domain

In order to evaluate the feasibility of utilising heuristic classification knowledge for browsing documents using an FCA lattice structure, it was only necessary to generate a single complete lattice for any formal context and gather statistical results about that generated lattice structure. However, the lack of available system resources and the significant quantity of documents for a single domain posed a problem for lattice generation. It was too time consuming to generate a complete concept lattice using the full set of documents. For this reason, iWeb FCA included a function that reduced the number of documents stored in all folders in the storage folder structure to contain, at a maximum, a specified amount. At a minimum, a folder could contain zero documents. Note the fact that the actual number of *folders* is not reduced means that all of the heuristic classification knowledge is still utilised to generate the complete concept lattice. This is because the MCRDR rules apply to particular folders in the storage folder structure, and not particular documents. In other words, the conclusions of the MCRDR rules are folders.

3.2.2 Generating a Complete Lattice

The batch process utilised to build the formal concepts and the concept lattice is an implementation of the general methodology of FCA for formulating concepts and building the concept lattice. The algorithm used in iWeb FCA was based upon the explanations of FCA provided by Richards (1998), Kim and Compton (2000), and Kim (2003). In detailing the procedure, C represents the formal context stored in iWeb FCA's database, D represents the set of objects (documents) in C, and M represents the set of attributes (rule keywords or folder names) in C. The procedure implemented is detailed in Figure 2.

Step 1:

Formulate an extent containing the set of objects G representing the largest concept of C. Then perform step 2 for each attribute m in the set M.

Step 2:

a) Find the set of objects X that contains the attribute m.

b) Check whether any previously formulated extent is equivalent to X.

c) If an equivalent extent of X does not exist, then add the set X as an extent of the attribute m.

d) Determine the intersection of X with all extents calculated in previous steps. If the intersection set does not exist, then add the intersection set as an extent of attribute m.

Step 3:

For each formulated extent, determine its intent:

$$Y \leftarrow \{\, m \quad M \mid (g,m) \quad I \text{ for all } g \quad X\}$$

Step 4:

*Construct the concept lattice by finding all the hierarchical **subconcept - superconcept** relations between all the formal concepts of C that were computed in steps 1 to 3.*

Figure 2 – Procedure for Generation a Concept Lattice in iWeb FCA

3.2.3 Browsing the Concept Lattice

A sample of the concept lattice browsing interface used in iWeb FCA is shown in Figure 3. As in the approach of Kim and Compton (2000), the lattice display is simplified by showing only direct neighbour nodes of the current node using hyperlinks. Each lattice node represents a concept comprised of a pair (X,Y), where X is the extent (a set of documents) and Y is the intent (a set of classification rule keywords) of the concept. The intents of each concept are used for indexing the terms of the browsing structure.

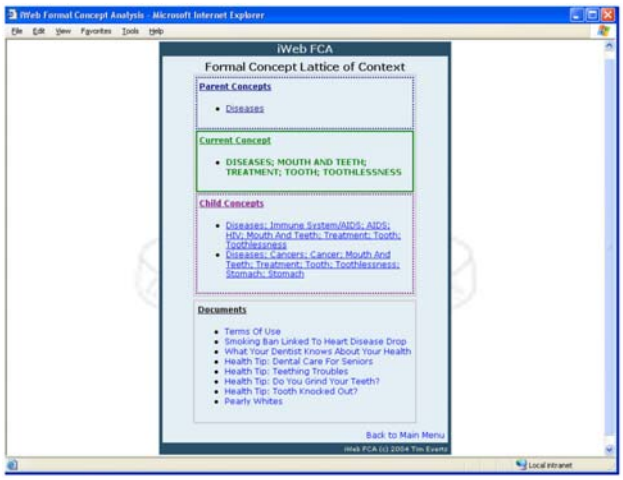

Figure 3 – iWeb FCA Concept Lattice Browsing Interface

The concept lattice browsing interface in iWeb FCA is divided into four distinctly recognisable sections. The current lattice node is displayed in green in a section labelled 'Current Concept', while parent nodes and child nodes are listed as hypertext links in sections labelled 'Parent Concepts' and 'Child Concepts' respectively. The set of documents associated with the current node are listed as hypertext links in a section labelled 'Documents'. The actual browsing of the lattice begins from the root node (concept) and the relationships of concepts can be explored by traversing from vertex to vertex by clicking on a child or parent node hypertext link. Each time a new node is selected, the interface is updated to show the parent and child nodes of the current node. The list of documents associated with the current node is also refreshed. Documents at a node can be viewed by clicking the appropriate hypertext link and the document will be displayed in a new Web browser window.

4 Evaluation

4.1 Data Set

The MCRDR heuristic classification knowledge utilised in this research study was collected over a period of time during a project undertaken at the University of Tasmania in Hobart, Australia. Table 1 summarises the data created as a result of the Web monitoring project which was focused on the domain of e-Health. In total, 7 sites were monitored by WebMon and 7588 documents were retrieved from those sites. Of those 7588 documents, 4598 were classified to the storage folder structure which contained 119 folders. During the classification process, 172 rules were created and a total of 285 unique rule conditions (keywords) were contained in those rules. The iWeb Web Portal site divided the complete storage folder structure into various sub-domains of eHealth, based on the individual folders at the second level of the storage folder structure. These sub-domains included 'Diseases', 'Demographic Groups', 'Drug Information' and 'Health and Wellness'. Dividing the complete storage folder structure into smaller parts simplified browsing for information, especially since the entire storage folder structure was quite large and the quantity of information was significant.

Web Monitoring	
Total Sites Monitored	7
Total Articles Collected	7588
Total Articles Classified	4598
Classification Knowledge	
Total Rule Used	172
Total Rule Conditions	285
Storage Folder Structure	
Total Folders	119

Table 1 – Summary of Web Monitoring Project

To conduct evaluation, a sub-domain of the eHealth domain was first selected to be utilised as the source of data for generating the concept lattice. The reason why only a sub-domain was selected is because the limited system resources available meant it would take a significant amount of time to generate a single complete concept lattice for the entire eHealth domain. Also, since the storage folder structure could be distinctly divided into the various sub-domains of eHealth (as is done on the iWeb Web portal site), it was much simpler to just deal with a small portion of the overall structure for the purpose of analysing it. Consequently, the sub-domain of 'Diseases' was selected for the purpose of the analysis. It contained the most information out of all the sub-domains and also had the largest storage folder structure. To enable a concept lattice to be generated from the Diseases sub-domain data, iWeb FCA was used to reduce the number of documents in any folder to be no more than 32. This figure was chosen through a trial and error approach based on the amount of time it took to generate a concept lattice with the available system resources. It resulted in a total number of 1063 classified documents making up the reduced data set.

4.2 Method

Having reduced the source domain data to a manageable amount for lattice generation, iWeb FCA was used to generate two different types of concept lattices. The first concept lattice was generated based on the MCRDR heuristic classification knowledge, and the second concept lattice was generated based on a combination of MCRDR heuristic classification knowledge (rule keywords) and abstract domain knowledge (folder names) because many of the folder names used in abstract domain knowledge also occur as keywords in the heuristic classification knowledge. For this reason, it may also be potentially useful to browse documents using a combination of the two knowledge types, especially because often a user does not make a clear distinction between the two knowledge types. Therefore, browsing a concept lattice based on this combination of knowledge types was also assessed as part of the evaluation undertaken.

The final step of the evaluation procedure was to gather and record statistics on the different browsing structures. This was done in order to assess the feasibility of utilising heuristic classification knowledge for browsing documents. Three main forms of analysis were performed. Firstly, the physical composition of the different browsing structures was analysed as a means of assessing the implications that each would have on browsing for documents. Secondly, the distribution of documents in the browsing structures was compared to determine whether utilising heuristic classification knowledge as a resource for browsing enhances a user's ability to locate a particular document. Finally, an analysis was performed on how the structures would actually be browsed. This final analysis was achieved by programmatically simulating the browsing process and recording information about each level that would be traversed in each browsing structure. The results and discussion of the analytical evaluation are presented in the Section that follows.

5 Result

5.1 Physical Browsing Structures

Table 2 shows the main statistics gathered from analysing the physical composition of the storage folder structure (SFS). Table 3 shows the statistics gathered from analysing the physical composition of a concept lattice which was generated based on the MCRDR heuristic classification knowledge (HCK lattice), as well as statistics for a second concept lattice generated on a combination of MCRDR heuristic classification knowledge and abstract domain knowledge (HCK-ADK lattice).

Total Number of Folder	80
Folders with Documents	56
Folders without Documents	24
Average Sub-Folders per Folder (without leaf folders)	6.08
Total Rules Utilised	78
Total Rule Keywords	109

Table 2 – Summary of Storage Folder Structure

	HCK	HCK-ADK
Total Number of Nodes (Concept)	77	88
Total Nodes with Documents	76	87
Total Nodes without Documents	1	1
Number of Single Level Nodes	22	3
Average Child Nodes per Node	1.69	1.69
Average Attributes per Node	4.08	7.18

Table 3 – Summary of Concept Lattice Structure

By comparing the physical composition of the SFS (see Table 2) with the HCK and HCK-ADK concept lattice structure (see Table 3), the implications of browsing documents based on heuristic classification knowledge as opposed to abstract domain knowledge can be made clear. In the SFS there is an average of 6.08 sub-folders for every folder (excluding leaf folders), while in the HCK and HCK-ADK lattice there is an average of 1.69 children nodes per node. Since the SFS is a hierarchical tree structure, it would be traversed starting from the root folder and finishing at a leaf folder. This means that in browsing the SFS a user tries to pick the best sub-folder at each step in order to locate a particular document. Each time a document is not located in a particular folder, the user would have to make the decision between an average of about 6 sub-folders as to where to go next. This also means that if a leaf folder is reached, it is difficult to know what to do next because the best guesses have already been made at each decision point.

However, with the HCK and HCK-ADK lattice structure, making the decision of where to go next is much less overwhelming for the user. This is because on average there is only about 1 or 2 child nodes to choose from. Also, since the HCK and HCK-ADK lattice is more of a network type structure, it means that if a document is not located by taking one path, it is possible to go back up another path rather than starting again. This opens up new decisions which have not previously been considered.

A further interesting aspect of utilising the HCK and HCK-ADK lattice for browsing documents is that every node except one (which would be the bottom-most node) contains at least one document (see Table 3). However, in the SFS there are 24 folders that do not contain any documents (see Table 2). This means there are 24 possible decisions a user could make when browsing the SFS that are potentially useless in locating a particular document. This not only makes locating a document more difficult in the SFS, but it would no doubt also increase a user's frustration.

Comparing the physical structure of the HCK-ADK lattice with the structure of the HCK lattice (Table 3) produces some very interesting results. The most interesting result is the significant decrease in the amount of single level nodes in the HCK-ADK lattice. In this analysis, a single level node is a node that has the *supremum* node (top most concept in the lattice) as its only predecessor, and the *infimum* node (bottom most concept in the lattice) as its only successor. If a large percentage of the total nodes in a lattice are single level nodes, it implies that the overall lattice structure is very shallow, meaning that more of the concepts will be general in nature. In regards to browsing the lattice for documents, this implies it will be more difficult for a user to locate the document desired. This is because there are fewer concepts in the lattice that would be specific enough to uniquely represent the attributes of that document.

Calculating the percentage of single level nodes in each lattice generated reveals that even though the HCK-ADK lattice contains 10 extra nodes (88 nodes) than the HCK lattice (77 nodes), only about 3 percent of nodes in the HCK-ADK lattice are single level nodes. However, in the HCK lattice, about 29 percent of all nodes are single level nodes. This implies that it would be much easier to locate a particular document when browsing the HCK-ADK lattice because a larger number of terms are being used to represent the attributes of documents resulting in a greater number of more specific concepts being generated.

5.2 Distribution of Documents

A second statistical analysis was undertaken to analyse how documents were distributed in the various browsing structures. The aim of this analysis was to determine whether utilising heuristic classification knowledge as a resource for browsing enhances a user's ability to locate a particular document.

The most significant result from analysing the distribution of documents in the SFS shows that the majority of the total 1063 classified documents are only located in a single folder. This implies that it would be quite difficult to locate a particular document when browsing the SFS because few documents can be found in multiple folders. Consequently, this makes the decision of which folders a user selects in searching for a document a lot more critical, since the likelihood of finding the document in a particular folder is relatively small.

The ability to locate a document can be significantly improved if the heuristic classification knowledge and abstract domain knowledge are used as a resource for browsing instead. In the HCK and HCK-ADK lattice, documents are distributed much more evenly than in the SFS. As a result, a larger amount of documents are located at a higher number of multiple locations (nodes) in the HCK and HCK-ADK lattice. This is also evident when the distribution of documents between the SFS, HCK and HCK-ADK lattice are compared graphically, as shown in Figure 4.

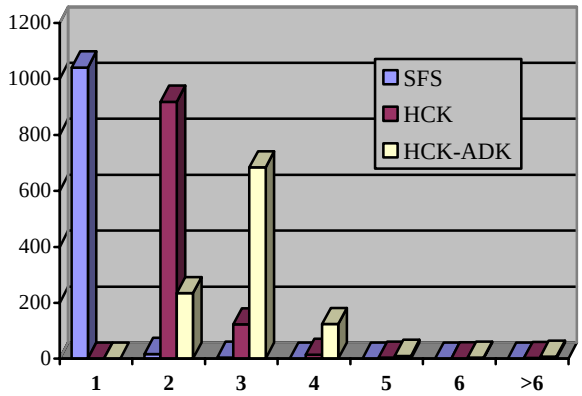

Figure 4 – Distribution of Documents in Multiple Locations

It is interesting to note the effect that utilising the terms from both knowledge types has on the distribution of documents in the lattice structures. In the HCK-ADK lattice, the distribution of documents appears to be more evenly spread than in the HCK lattice. This can be clearly seen in Figure 4. Also, in the HCK-ADK lattice, 78 percent of documents are located at 3 or more nodes, whereas only about 14 percent are located at that many nodes in the HCK lattice. This shows that the utilisation of the terms of both knowledge types can also provide more possibilities for locating a document while browsing.

5.3 Analysis of Browsing

The final statistical analysis undertaken involved simulating the way a user might actually browse each of the different structures. For the storage folder structure (SFS) this was simulated programmatically by beginning at the first level of browsing, namely the root folder and recording information about the properties of that browsing level. Then the entire SFS was traversed one level (folder) deeper to all sub-folders visible from the first level, and the properties of that level were also recorded. This process continued until it was not possible to traverse any deeper, namely when all folders on the browsing level were leaf folders.

A similar programmatic simulation was also applied to the generated concept lattices to record the information about each level of browsing in the lattice structure. The deepest level of browsing in the lattice was the level that contained only the *infimum* node (bottom most concept in the lattice). It should be noted that the structure of a concept lattice is such, that when browsing the lattice an individual node may appear (be visible) at two different browsing depths, depending on which path is taken through the lattice.

The statistics that were recorded at each level of browsing included the total number of folders or nodes for that level, the total number of documents, the total number of unique documents, and the average number of documents per folder or node on that level.

(a) Storage Folder Structure

Browsing Depth (folders)	Total Folders	Total Docs	Unique Docs	Average Docs per Folder
1 Level	1	0	0	0.00
2 Level	20	489	487	24.45
3 Level	59	602	586	10.20

(b) HCK Concept Lattice

Browsing Depth (Nodes)	Total Nodes	Total Docs	Unique Docs	Average Docs per Node
1 Level	1	1063	1063	1063.0
2 Level	46	1088	1063	23.65
3 Level	25	152	145	6.08
4 Level	6	9	7	1.50
5 Level	1	2	2	2.0
6 Level	1	0	0	0.00

(c) HCK-ADK Concept Lattice

Browsing Depth (Nodes)	Total Nodes	Total Docs	Unique Docs	Average Docs per Node
1 Level	1	1063	1063	1063.00
2 Level	21	1166	1063	55.52
3 Level	46	852	829	18.52
4 Level	20	68	60	3.40
5 Level	5	8	6	1.60
6 Level	1	2	2	2.00
7 Level	1	0	0	0.00

Table 4 – Analysis of Browsing

Table 4 presents the statistics gathered by simulating browsing the storage folder structure (SFS), the HCK lattice and the HCK-ADK lattice.

The first and perhaps most obvious comparison that can be made between the SFS and HCK and HCK-ADK lattice is the difference in the number of browsing levels. Starting at the root folder (level 1) in the SFS, it is possible to traverse to a maximum browsing depth of 3 levels. On the other hand, in the HCK lattice it is possible to traverse to a maximum browsing depth of 6 levels and in the HCK-ADK lattice to 7 levels.

The SFS appears to be much easier for a user to browse because there are fewer levels of browsing in it. However, the fact that there are fewer levels of browsing means that the amount of folders on each level is quite large. The structure of the SFS is such, that the deeper the user browses, the larger the amount of folders that appear on each level. This means the decision of which folder to select when trying to locate a document becomes much more difficult with each new level that is traversed. In the HCK lattice the opposite is the case. Disregarding the first level of browsing (the root node), the deeper a user browses the HCK lattice structure, the fewer the nodes that appear at each browsing level. Therefore the decision of where to go next when browsing the HCK lattice only becomes easier rather than more difficult.

It is also interesting to compare the total number of documents and unique documents at each level of browsing in the SFS and the HCK / HCK-ADK lattice (see Table 4). Since the SFS only has three levels, it is appropriate to compare only the first three levels of both structures. This comparison reveals that all 1063 classified documents can be located at both of the first two levels of browsing in the HCK / HCK-ADK lattice, while not even half of all the documents can be found at each of the same two levels of browsing in the SFS. This would suggest that there is more chance of locating a desired document in the HCK / HCK-ADK lattice as there is in the SFS.

Comparing the difference between the HCK-ADK lattice and the HCK lattice shows that there is only one extra level of browsing in the HCK-ADK lattice. Another interesting statistic is that the average number of documents per node on nearly all the levels of browsing in the HCK-ADK lattice is significantly higher than that in the HCK lattice. Furthermore, the overall difference between the number of total and unique documents on each level in the HDK-ADK lattice is also significantly higher than in the

HCK lattice. Therefore, from the comparisons presented it can be concluded that the utilisation of the terms of both knowledge types improves the possibility of locating a document during browsing. This makes the browsing experience all the more beneficial for a user.

6 Conclusion

The investigation undertaken in this study was aimed at determining the feasibility of utilising heuristic classification knowledge acquired through the use of MCRDR as a resource for browsing documents retrieved from the WWW. A Web-based system was developed which generated a FCA concept lattice using the heuristic classification knowledge of MCRDR. To evaluate the feasibility of utilising heuristic classification knowledge as a resource for browsing documents, a comparative statistical analysis was performed. This involved comparing the difference between browsing documents using two different structures. Namely, a storage folder structure (SFS) based on abstract knowledge of a domain, and a concept lattice based on MCRDR heuristic classification knowledge.

From the evaluation performed, it is concluded that the concept lattice-based browsing scheme of FCA provides a feasible way to utilise MCRDR heuristic classification knowledge for browsing documents of a specific domain. An analysis of the physical composition of the SFS compared with the concept lattice structure revealed that browsing based on heuristic classification knowledge significantly simplifies each decision a user has to make during browsing. Also, analysing the distribution of documents in each browsing structure revealed that a user's ability to locate a particular document when browsing the lattice structure is significantly enhanced. Documents are more evenly distributed throughout the lattice than in the SFS, and they can also be found in a larger number of multiple locations. Furthermore, by programmatically simulating the way a user might browse each structure, it was possible to determine the options they would be presented with during browsing. Even though the lattice structure based on heuristic classification knowledge appeared to require more interaction from a user during browsing than when using the SFS, the browsing experience is much less overwhelming because each individual stage of browsing is much simpler.

In addition, the results of a secondary investigation concluded that using the terms of both abstract domain knowledge and heuristic classification knowledge also presents itself as a viable option for browsing documents. Statistically comparing a lattice generated on the terms of both knowledge types with a lattice generated plainly on heuristic classification knowledge produced some interesting results. The results showed that the utilisation of the terms of both knowledge types provides a much richer context for browsing. Each document can not only be found at a larger number of multiple locations in the lattice, but the extra terms also enable the location of each document to be identified more specifically.

7 Further Work

There are potentially several areas of research related to this study that can be investigated. An immediate continuation of the work undertaken might be to incorporate the prototyped concept lattice browsing approach of iWeb FCA into the iWeb Web Portal Site. This may be useful for providing an alternate method to users for browsing documents on that site, especially considering the significant quantity of information available.

An aspect that was not covered by this study is a user's actual satisfaction of browsing documents based on heuristic classification knowledge, as compared with browsing based on abstract domain knowledge. To evaluate this would also be interesting and would most likely involve performing a quantitative user study. The study could compare and assess the performance of browsing documents based on each type of knowledge.

It may also be interesting to investigate the use of other classification knowledge types as a resource for browsing documents. This study simply utilised the classification knowledge of MCRDR because it was readily available and suitable. There may well be other types of classification knowledge that can be utilised appropriately for browsing documents. In the same manner, it may also be useful to evaluate the use of an alternate browsing structure, other than the concept lattice of FCA, that can also utilise heuristic classification knowledge as a resource for browsing documents.

However, perhaps the most interesting point that remains to be seen is whether browsing schemes based on heuristic classification knowledge will become a standard for browsing information on the WWW. With the consistent increase in the amount of information being generated on the WWW, there is an increasing need for more effective and simple ways of locating and retrieving information. To this extent, the utilisation of heuristic classification knowledge as a resource for browsing and searching of information may provide a potential solution to this problem.

8 References

Becker, P. (2005). "Using intermediate representation systems to interact with concept lattices." Formal Concept Analysis. Third International Conference, ICFCA 2005. Proceedings (Lecture Notes in Artificial Intelligence Vol.3403): 265-268.

Boyapati, V., K. Chevrier, et al. (2002). ChangeDetector[tm]: a site-level monitoring tool for the WWW. WWW 2002.

Carpineto, C. and G. Romano (2005). Formal Concept Analysis as Mathematical Theory of Concepts and Concept Hierarchies. Formal Concept Analysis Foundations and Applications. B. Ganter, G. Stumme and R. Wille.

Cole, R. J. (2000). The Management and Visualisation of Document Collections Using Formal Concept Analysis, Griffith University**:** 122.

Cole, R. J., P. W. Eklund, et al. (2004). Browsing Semi-Structured Texts on the Web Using Formal Concept Analysis. Intelligent Technologies for Information Analysis. N. Zong and J. Liu, Springer**:** 243-264.

Correia, J. H., G. Willie, et al. (2003). "Concpetual Knowledge Discovery - A Human-Centered approach." Applied Artificial Intelligence **17**: 281-302.

Dazeley, R. and B. Kang (2003). Weighted MCRDR:Deriving Information about Relationships between Classifications in MCRDR. 16th Australian Joint Conference on Artificial Intelligence (AI'03), Perth, Australia.

Eklund, P. and B. Wormuth (2005). "Restructuring help systems using formal concept analysis." Formal Concept Analysis. Third International Conference, ICFCA 2005. Proceedings (Lecture Notes in Artificial Intelligence Vol.3403): 129-144.

Ganter, B., G. Stumme, et al. (2005). Formal Concept Analysis Foundations and Applications.

Garter, B. and R. Willie (1997). "Applied Lattice Theory:Formal Concept Analysis."

Kang, B., K. Yoshida, et al. (1997). "Help desk system with intelligent interface." Applied Artificial Intelligence **11**(7-8): 611-631.

Kim, M. (2003). Document Management and Retrieval for Specialised Domains: An Evolutionary User-Based Approach, University of New South Wales.

Kim, M. and P. Compton (2000). Developing a domain-specific Information Retrieval Mechanism. 6th Pacific Knowledge Acquisition Workshop (PKAW 2000), Sydney Australia.

Kim, M. and P. Compton (2001). Formal Concept Analysis for Domain-Specific Document Retrieval Systems,. 13th Australian Joint Conference on Artificial Intelligence (AI'01), Adelaide Australia, Springer-Verlag.

Kim, M. and P. Compton (2004). "Evolutionary document management and retrieval for specialized domains on the web." International Journal of Human-Computer Studies **60**(2): 201 - 241.

Kim, Y. S., S. S. Park, et al. (2004). Adaptive Web Document Classification with MCRDR. International Conference on Information Technology: Coding and Computing ITCC 2004, Orleans, Las Vegas, Nevada, USA.

Liu, L., C. Pu, et al. (2000). WebCQ: Detecting and Delivering Information Changes on the Web. International Conference on Information and Knowledge Management (CIKM), Washington D.C., ACM Press.

Liu, L., W. Tang, et al. (2002). "Information Monitoring on the Web:A Scalable Solution." World Wide Web Journal **5**(4).

Mladenic, D. (1999). "Text-learning and Related Intelligent Agents." Applications of Intelligent Information Retrieval.

Park, S. S., S. K. Kim, et al. (2003). Web Information Management System: Personalization and Generalization. the IADIS International Conference WWW/Internet 2003.

Park, S. S., Y. S. Kim, et al. (2004). Web Document Classification: Managing Context Change. IADIS International Conference WWW/Internet 2004, Madrid, Spain.

Quan, T. T., S. C. Hui, et al. (2005). "A fuzzy FCA-based approach for citation-based document retrieval." 2004 IEEE Conference on Cybernetics and Intelligent Systems (IEEE Cat.04EX912): 578-83 vol.1.

Rajapakse, R. K. and M. Denham (2003). A Reinforcement Strategy for (Formal) Concept and Keyword Weight Learning for Adaptive Information Retrieval. MLIRUM'03: Second Workshop on Machine Learning, Information Retrieval and User Modeling at the Ninth International Conference on User Modeling, Pittsburgh, PA, USA.

Richards, D. (1998). The Reuse in Ripple Down Rules Knowledge Based Systems, University of New South Wales.

Richards, D. (2001). "Combining cases and rules to provide contextualised knowledge based systems." Modeling and Using Context. Third International and Interdisciplinary Conference, CONTEXT 2001. Proceedings (Lecture Notes in Artificial Intelligence Vol.2116): 465-469.

Sebastiani, F. (2002). "Machine learning in automated text categorization." ACM Computing Surveys **34**(1): 1-47.

Tang, W., L. Liu, et al. (2000). WebCQ Detecting and Delivering Information Changes on the Web. Proc. Int. Conf. on Information and Knowledge Management (CIKM).