Improvement of Precision and Recall for Information Retrieval in a Narrow Domain: Reuse of Concepts by Formal Concept Analysis

W.C. Cho, D. Richards
Department of Computing
Macquarie University
Sydney, Australia
{wccho, richards}@ics.mq.edu.au

Abstract

With the exponential growth of the World Wide Web (WWW), it has become the most popular place to gather information. However the size of the WWW makes it difficult for people to locate relevant information. About 85 % of all Web users use search engines of some kind for this purpose. However, existing search engines often do not return relevant information.

The main focus of this paper is to improve search performance by using keywords and web pages which have been previously used or visited by other users. The Formal Concept Analysis (FCA) method has been adapted to maintain a concept map of keywords. This paper shows how both precision[1] and recall[2] has improved the specific-domain area, in which users can share the same knowledge.

1. Introduction

Both the number of users and the amount of information available has exploded at a remarkable rate since the advent of the World Wide Web (WWW) [10][14]. 85% of all Web users use various search engines and search tools to find relevant Web sites for specific information [4][11]. Web search engines generally require the requested information to be immediately available, accurate and suit the users' requests [12].

The purpose of a search engine is to provide information in response to a user's query. This information should not only be accurate but related to the user's domain. Unfortunately, current search engines generally return a high level of ambiguous information. Many Web users have been dissatisfied with using search engines. The main reasons for dissatisfaction are slow access, the inability to find relevant information and broken links (*percentage of dead-pages: Northern Light 5.7%, Google 4.3%, AltaVista 13.7%, MSN Inktomi 2.6%, Anzwers 1.3%, HotBot 2.3%, Fast 2.3%*)[3]. These poor retrieval results depended on where the information was cited from [11]. The ultimate goal of search engines is to recall information that is both accurate and relevant. Existing search engines on the Web have incorporated a number of techniques to assist in both the recall and precision of information. Although every effort has been made in trying to accomplish this, there is one underlying factor. Search engines treat a user's query as ASCII text without any regard to the meaning of the query. This also includes the search for information to a query across multiple domains.

Users share domain-specific knowledge in a specific-domain area. Within a narrow domain, it is then possible to reuse the knowledge with a cache mechanism in the specific-domain, to provide the user with a related concept and location within a map using Formal Concept Analysis (FCA). The purpose of the work reported in this paper is to build a system for specific-domain area users, and to improve the relationship between precision and recall proportionately in a specific-domain area.

2. Methodology for the Proposed System

2.1. Formal Concept Analysis

Wille [15] developed Formal Concept Analysis (FCA). FCA is based on the understanding of a concept as an entity of thought, which consists of an extension and intension. Lattice theory is used for representing concepts [2][6]. FCA provides a substitute graphical representation of tabular data that is somewhat instinctive to navigate and use [7].

2.1.1 Formal Context with Classification of Concepts

The fundamental conceptual structure of FCA is the formal context (K). A formal context comprises a set of objects and their attributes. A formal context constitutes a triple (G, M, I). 'G' is the set of objects, 'M' is the set of

[1] Percentage of total relevant documents retrieved form all documents.
[2] Percentage of total relevant documents retrieved that the searcher is actually interested in.

[3] http://www.searchengineshowdown.com/stats/dead.shtml

attributes and 'I' is a binary relation defined between 'G' and 'M'. Their relation is represented as $I \subseteq G \times M$. Hence, we can define a formal context (K) as: K = (G, M, I). If an object *g* has an attribute m then $g \in G$ is related I to m which is indicated by the relationship $(g, m) \in I$ or *gIm*.

In our application of FCA Web-documents coincide with objects and the user's query (keywords) related to the Web-documents comprises attribute sets. We can suppose that the formal context of Web-documents (K) represents as: K = (G, M, I), and five objects (G) are such as: {Web-document1, Web-document2, Web-document3, Web-document4, Web-document5} and five attributes (M) are such as: {html, html tag, html font, html mobile, html rollover}. 'I' is represented such as: {(Web-document1, html), (Web-document1, html rollover), (Web-document2, html), (Web-document2, html tag), (Web-document3, html), (Web-document3, html tag), (Web-document3, html mobile), (Web-document4, html), (Web-document4, html font), (Web-document4, html mobile), (Web-document5, html), (Web-document5, html mobile)}. Table 1 shows the formal context for K where K is domain of queries related to developing web pages. In the table 1, 'X' indicates that an object has an attribute. For example, the object "Web-Doc1" has the attribute "Html" and "Html Rollover".

Table 1. A representation of formal context for the Web-documents. An 'X' indicates that an object has that attribute.

	Html	Html Tag	Html Font	Html Mobile	Html Rollover
Web-Doc1	X				X
Web-Doc2	X	X			
Web-Doc3	X	X		X	
Web-Doc4	X		X	X	
Web-Doc5	X			X	

The notion of a formal concept (G, M, I) is represented as a pair (A, B). 'A' is the set of objects ($A \subseteq G$) and 'B' is set of attributes ($B \subseteq M$). Its two operators are:

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

These two operators are used to formalize the notion of a formal context. The above operators, A' and B' are the set of objects and attributes respectively. A' is included by all objects in A. B' is included by all attributes in B. By finding all intersections of the primitive concepts given in the formal context we can generate concept and using the subsumption operator $\geq$ we order all concepts to form a concept lattice of all concepts (see Figure 1).

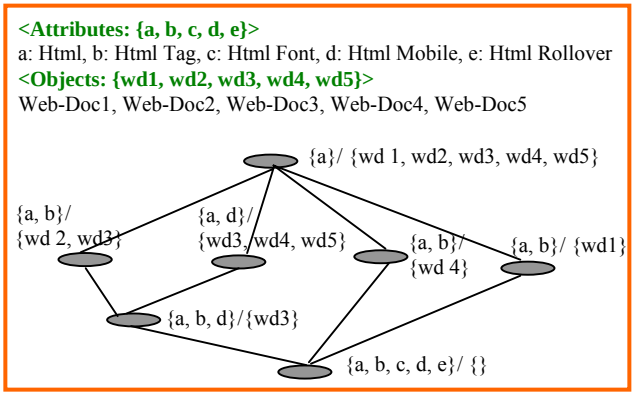

Figure 1. Concept lattice shows formal context in Table 1.

2.2. Cache Mechanism

Caching was originally introduced into computer architecture in order to increase speed and performance by migrating data copies from slow main memory to points closer to fast CPU. On the Web, caching was first implemented into client Web browsers [1]. Recently retrieved documents can be viewed faster from a client cache. This concept was extended to the network and led to development of a proxy cache [8]. Since many clients use the same cache for accessing the Web [5], it can serve requested objects faster after caching.

2.3. Meta Search Engine for Web Information Collection on the Web

Web information search engines allow users to query using their own built-in database. Unlike general search engines, however, Meta- search engines do not have to have their own database [3][13]. Instead, Meta- search engines send the user's query to other search engines simultaneously and provide the user with the results it has collected from other general search engines. Therefore, meta- search engines provide a useful and comprehensive way of finding information on the Web by combining several search engines.

3. Implementation

3.1. Specific-Domain Search Engine (SDSE)

Users share knowledge of a specific-domain in the same-domain or specific-organisation area. Therefore it is desirable to create an ontology for the given domain.

The Specific-Domain Search Engine (SDSE) system (see Figure 2) classifies user's queries by applying FCA

to develop a Concept Tree Map (CTM) in the form of a lattice. The information is saved by the cache mechanism. This system checks the CTM against the user's queries. Based on the query, the system provides the user with information but if the user decides the result is unacceptable, the system proposes a list of concepts related to the user's query.

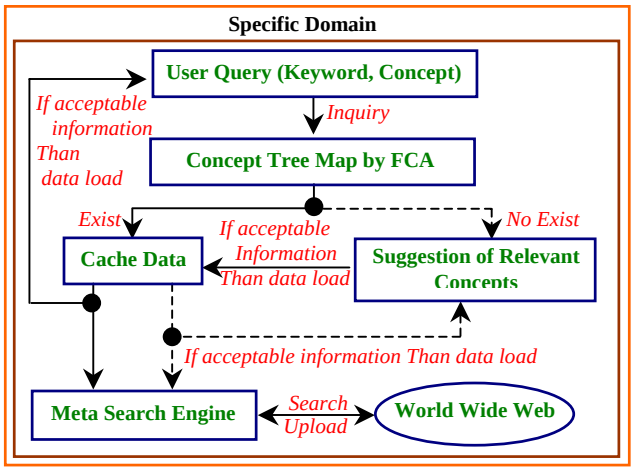

Figure 2. Diagram of SDSE System

However, if the user does not find any of the concepts to match their query or they need more information, the user can search (or collect) information for their own query through the Meta-search engine in this system. At this time, the user can save the new information and store it for use by others by adding a new rule.

3.1.1 Adding a new Rule: Classification of Concept and Representation by Lattice Model in FCA

An addition of a new rule (concept) into the SDSE system is performed by users and represented using the FCA lattice model. For example, if we assume to add a new concept (called "a b f") to a concept (called "a b c") using FCA we follow the process depicted in Figure 3.

- *Step 1:* If the SDSE system did not discover (or Find) any matching concept(s) regarding the new concept of the current user, the system provides suggestions related to that concept from the cached concepts in the system.

- *Step 2:* The new concept {a b f} is classified using FCA into seven concepts in the SDSE system. The seven concepts are: {a}, {b}, {f}, {a b}, {a f}, {b f}, {a b f}.

- *Step 3:* When new concepts are added, they are compared with the existing concepts ({a}, {b}, {c}, {a b}, {a c}, {b c} and {a b c}) using the FCA rules in the SDSE system. Through comparison we can determine that the three concepts {a}, {b} and {a b} already exist in the concept tree provided by the FCA lattice structure.

- *Step 4:* The new concepts {f}, {a f}, {b f} and {a b f} are added to the set of existing concepts except in the case of duplicate (overlap) concepts. The concept {f} is linked to {Root} and the concept {a f} is linked to the concept {a} and {f}. Therefore, the concept {a f} include attributes of both {a} and {f}.

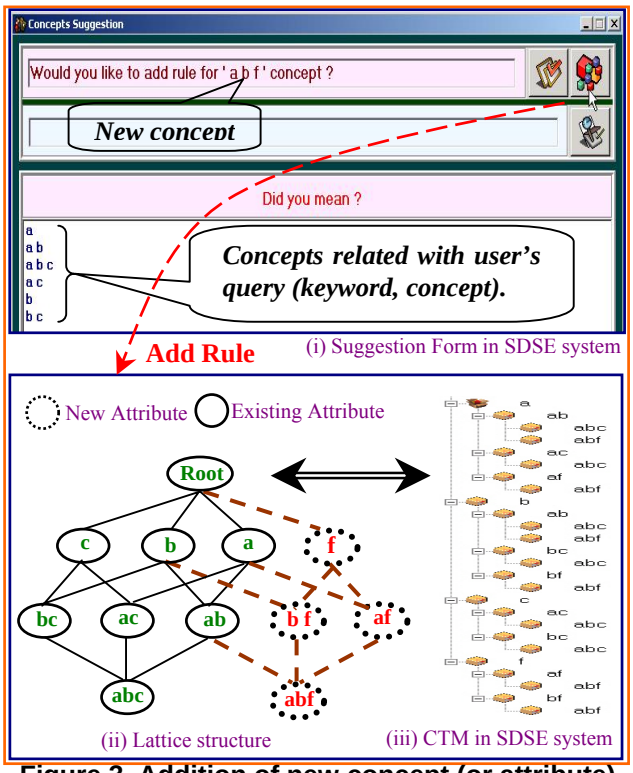

Figure 3. Addition of new concept (or attribute) and representation in SDSE

3.1.2 Reuse Knowledge

The SDSE system reuses domain-specific knowledge by using a cache mechanism. This reduces the search time against current search engines (Yahoo, Google and so on). It also improves accuracy of the user's query. In Figure 4, if one user needs information about 'html rollover', the user is able to access the information conveniently and easily through already cached (saved) data.

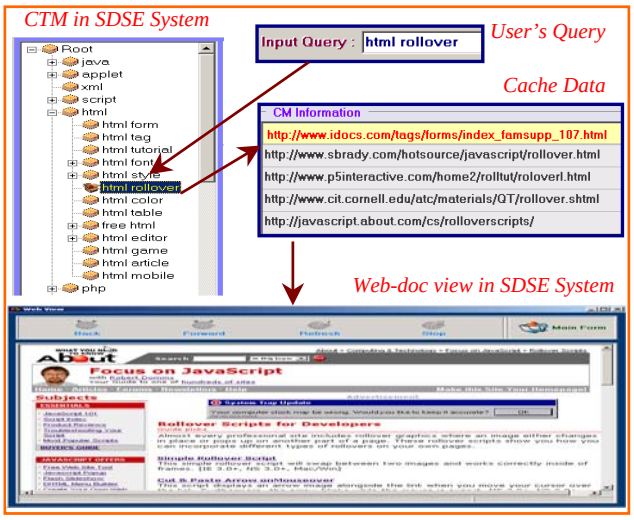

Figure 4. An example of Cache Mechanism in SDSE System

3.1.3 Navigation for Concepts

The SDSE system builds a map of a user's query by using FCA, which provides the concept and location of the relative query of a current user. The FCA map gives an ordinary (novice or expert) user a friendly hierarchical tree. Hence the user can extend the concept of users' query with two kinds of maps, enabling the user to make the correct decision based on well-coordinated information.

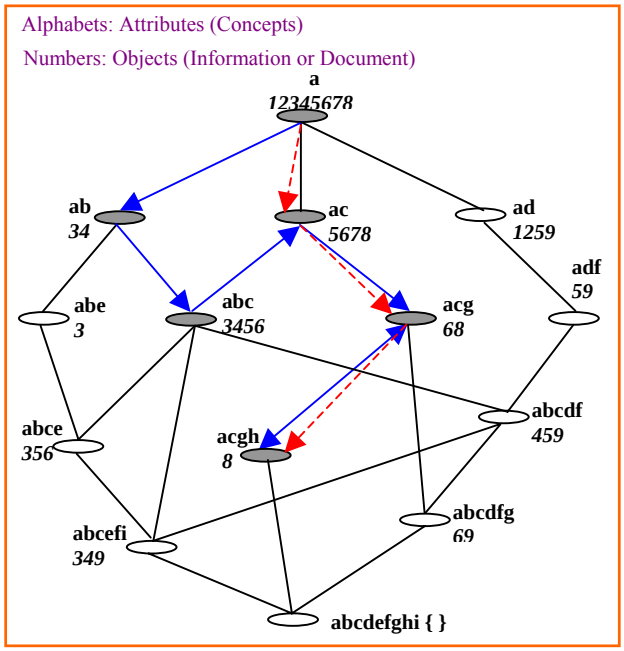

Figure 5. Navigation of Concept

This lattice structure has been drawn incrementally with paths from previous users. The lines show related information comparing their attributes. For example in Figure 5 [9], two users (X and Y) are seeking the information contained in document 8. User X may have followed the path ({a}→{ab}→{abc}→{ac}→{acg}→{acgh}) and user Y may have followed the path ({a}→{ac}→{acg}→{acgh}). Using the lattice we can shorten the search path and reduce the effort by showing the user attributes, such as ag, ac, acd so on. Further, by clicking the icon ( ) or enter-key, the user can obtain the search results showing the location of the current concept (query), the parents of the query and the child concepts of the query as shown in Figure 6. In this way the lattice generated by FCA provides a domain specific ontology which structures the domain and provides the context for the query.

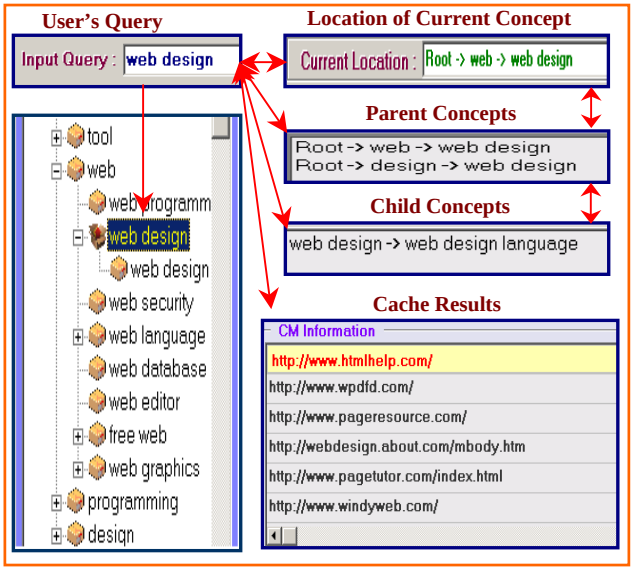

Figure 6. An example of navigating the concept in SDSE System

4. Experiment

4.1. Aims

The work described above is based on two assumptions:

- it is possible to improve precision and recall of user's query with a cache mechanism
- it is possible to improve precision and recall of user's query using a shared domain specific ontology.

We now describe an experiment we have conducted to test these assumptions.

4.2. Data Sets

For this experiment we specified that queries were to be restricted to the narrow topic area "How to make a homepage?". We describe next the groups involved in the study, the tasks and our results.

4.2.1. Groups

17 volunteers in total participated in this experience; divided into three groups S (6 persons), X (5 persons), Y (5 persons). 16 people consisted of computing students at the University of Tasmania. The 17th participant is an expert in this specific-domain and has been recruited to provide an oracle with more correct recall.

- *S Group:* This group is used to initialize the system. This group puts FCA rules into the system and caches information.
- *X Group:* This group is users to determine the precision and recall of the user's query in the fixed data in the system. Also it adds FCA rules into the system and caches collected information.
- *Y Group:* This group obtains the precision and recall of user's query whenever the rule and data is added into the system.

4.2.2. Query cases

The total number of query cases (episodes) for this experiment was 380. S group provided the initial cases with which the system was initialized in preparation for the experiment. S group included 6 students who generated 30 queries per person. X group had 5 students who used 20 queries each for a fixed rule experiment. The Y group had 5 students who used 20 queries for the expansion rule experiment. In total we have 180 queries created by group S, and 100 queries used by each of the X and Y groups from the original 180 cases. From the 180, 25 queries are the same. In 61 queries there are overlaps (*Html 2, Html form 2, Html tag 2, Html font 2, Icon 4, Perl 4, Asp 3, Asp tutorial 2, Cgi 5, Jsp 3, Jsp tutorial 2, Javascript 5, Svg 2, Css 2, Dhtml 4, Flash 3, Xml 3, Cookies 2, Rollover 2, Php 3, Php tutorial 2, Photoshop tutorial 3, Web design 2, Web palette 2, Image map 2*) among the total collection of 180. Hence, for this experiment, the actual used initial cases are 144 queries (180-(61-25) = 144).

Table 2. Query Case for Experiment

Description	Group	Total Number	Actual Number
Initial Setup	S	180	144
Query of Same Name	S	25	
Overlap of Same Query	S	61	
Experiment for Fixed Rule	X	100	100
Experiment for Extension Rule	Y	100	100

4.3. Experimental Progress

Using the following steps, the experimental progress is as shown Figure 7.

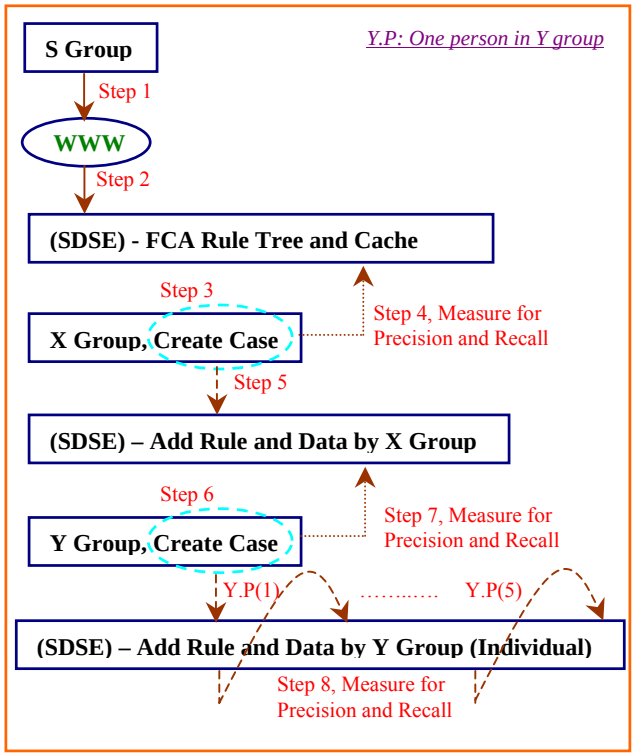

Figure 7. Diagram of Experiment

Step1: The S group with 6 people created 30 query cases each in the given specific domain. And then, each person finds the reasonable information results of each query with common general search engine (yahoo, google, hotbot and so on.).

Step2: Initialize the SDSE system with queries and information, which is acquired by S group.

Step3: Each member of the X group tests 20-query cases (from the initial 180) in the specific domain area with the SDSE system. In this step, rules and data are not added into system.

Step4: Measure recall and precision from X group.

Step5: Each person from X group adds rules and data into the SDSE system with the query cases used in step3.

Step6: Test precision and recall whenever each person from Y group generates 20 query cases and X group adds rules and data into the system. At this time, Y group doesn't input rules and data into the system.

Step7: Measure recall and precision from Y group.

Step8: Each person from Y group adds rules and data into the SDSE system with query case used in step6. In each test, they measure precision and recall.

4.4. Results

Figure 8 presents the results of step4 in section 4.3. We can see that precision and recall are very steady without adding rule and information data of the query into the system. The Figure 9 presents the results of step7 in section 4.3. In Figure 10, there is an improvement of precision and recall when user adds the rule and information data of the query into the system. Therefore, we could confirm two interesting results come out in this experiment. Firstly, compared fixed rules and flexible (additive) rule, when fixed rules are using, the recall and precision do not increase and decrease very steady (see Figure 8). However, when additive rules are using, the recall and precision increased with each 4.64% and 4.68% (see Figure 9). Lastly, Y group tested adding rules individually. Therefore, the recall is average 2.1 % increased and precision is average 3.4% increased in experimental system (see Figure 10).

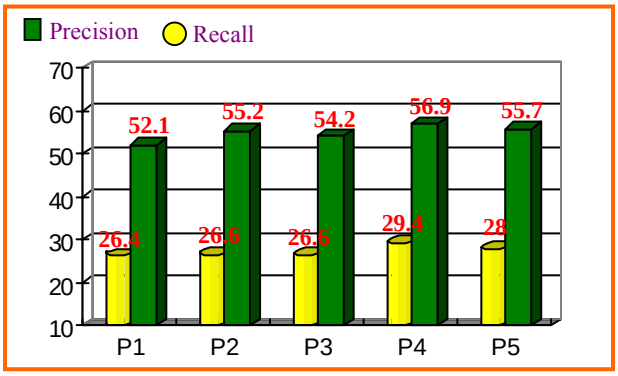

Figure 8. Result of experiment against fixed rules and data in step 4 of section 4.3.

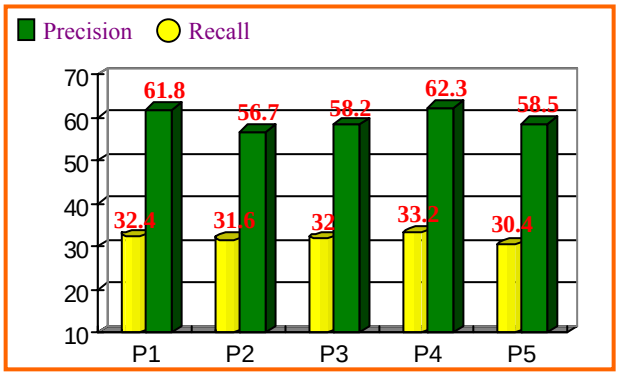

Figure 9. Result of experiment against added rules and data by X group (Step 7 of section 4.3.)

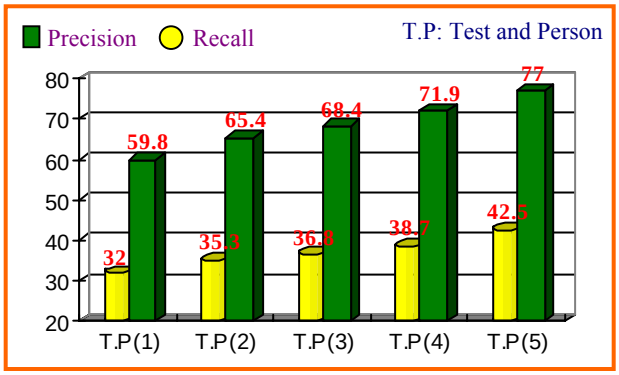

Figure 10. Result of experiment against added rules and data by individual testing in Y group (Step 8 of section 4.3.)

This experiment has demonstrated that the system provides a convenient method and retrieves more accurate information by using a cache mechanism and concept navigation map in a specific-domain. These results show that the system increases precision and recall of user's query with information gathering using meta-retrieval engine.

5. Conclusions and Future Work

The main purpose of this work is to improve recall and precision synchronously. We combined three methods, such as FCA, cache mechanism, and Meta search engines to achieve our purpose. FCA has been adapted to maintain a concept map, which assists users to find a suitable query and to manage the concept map easily. Therefore, the incorporation of the FCA algorithm has resulted in reduced ambiguity in the user's query. This assists users to find exact concepts in response to their queries. The cache mechanism presents the users with results more quickly, virtually instantly. This is achieved by the cache mechanism reducing the time to load Web documents as well as stopping the delivery of dead

(broken) Web documents. Hence, cache mechanism promotes efficiency of precision of Web document retrieval. The Meta search engine provides more valuable web information.

We developed a prototype system within the specific-domain, "How to make a home page?" The system has been integrated with these methods and we have performed evaluation of the system with users. We were able to achieve improvements in recall and precision concurrently.

Our current and future work is focused in two areas. Firstly, we should consider the hit rate and sequence times within a certain period and delete old or rarely used cache data. Deletion of obsolete data may increase cache mechanism performance. Finally, we are interested in finding means to extract concepts automatically and to discover the relationships between concepts in the same domain.

References

[1] D. Barnes and N. Smith, "An analysis World Wide Web Proxy Cache Performance and its application to the modeling and simulation of network traffic," *Proceedings of the 4th International Conference on Telecommunication Systems*, 1996.

[2] G. Birkhoff, *Lattice Theory 3rd edition*, American Mathematical Society, Incremental Clustering for Dynamic Information Processing, ACM Transactions on Information Processing Systems, 11, 1993, pp. 143-164.

[3] D. Dreilinger "Experiences with Selecting Search Engines using Meta-Search," *ACM Transaction on Information Systems*, 1997.

[4] G.E. Dupret and M. Kobayashi "Information Retrieval and Ranking on the Web: Benchmarking studies I," *IBM TRL Research Report*, 1999.

[5] L. Fan, P. Cao, J. Almeida and A. Broder, "Summary Cache: A scalable Wide-Area Web Cache Sharing Protocol," *IEEE ACM Transactions on Networking,* 2000.

[6] B. Ganter and R. Wille, *General lattice theory 2nd edition*. Birkhauser Verlag, Basel, 1998, pp. 591-605.

[7] B. Ganter and R. Wille, "Formal Concept Analysis – mathematical Foundations Berlin: Springer Verlag," 1999.

[8] J. Gwertzman, "World Wide Web Cache Consistency." *USENIX Annul Technical Conference*, 1996.

[9] M.H. Kim and P. Compton, "Developing a Domain-Specific Document Retrieval Mechanism," *Proceedings of the 6th Pacific Knowledge Acquisition Workshop*, Sydney, Australia, December 11-13, 2000, pp.189-206.

[10] O. King and M. Kobayashi, "Information Retrieval and Ranking on the Web: Benchmarking studies II", 1999.

[11] M. Kobayashi and K. Takeda, "Information Retrieval on the Web," *IBM Research*, 2000.

[12] S.Lawrence, "Context in Web Search," *IEEE Data Engineering Bulletin*, Volume 23, Number 3, 2000, pp. 25-32.

[13] S. Lawrence and C.L. Giles, "Context and page Analysis for Improved Web Search," *IEEE Internet computing*, 1998.

[14] I. Melve, "Web Caching Architecture," *DESIRE Web caching team*, 2001.

[15] R. Wille, "Restructuring lattice theory: an approach based on hierarchies of concepts," *Ordered sets*, 1982, pp.445-470.