A Reinforcement Learning Strategy for (formal) Concept and Keyword Weight Learning for Adaptive Information Retrieval

Rajapakse, R.K. and Denham, M.

Centre for Neural and Adaptive Systems, University of Plymouth, UK
{rohan,mike}@soc.plym.ac.uk

Abstract. This paper reports our experimental investigation into the use of a reinforcement learning strategy to learn weights of (formal) concepts and keywords to support Information Retrieval. This work is a part of our main research objective of using more elegant construct of a concept rather than simple keywords as the basic element of representation and matching. The framework used for achieving this was based on the theory of Formal Concept Analysis (FCA) and Lattice theory. Features or concepts (formulated according to FCA) of each document (and query) are represented in a separate concept lattice and are weighted separately with respect to the document. The document retrieval process is viewed as a continuous conversation between queries and documents, during which documents are allowed to learn a consistent set of significant concepts to help their retrieval. The learning strategy used was based on relevance feedback information that makes the similarity of relevant documents stronger and nonrelevant documents weaker. Test results obtained on the Cranfield collection show a significant increase of average precisions as the system gains more experience.

1 Introduction

The human brain is unquestionably the best IR machine in terms of effectiveness. Unfortunately, the study of human brain functions is still in its infancy and how exactly the brain works is not clear yet. However, the superiority of human brain in IR tasks seems to come by three major properties of the brain.

1. its ability to read and understand the concepts, ideas or meanings central to the document
2. its ability to reason out the usefulness of documents to information needs (queries) based on the understanding of the concepts (ideas or meanings) gained by reading the contents of documents and queries, and
3. its learning capability to make us adaptive to the environment allowing us to gain knowledge through learning or interaction with the environment.

Understanding concepts, ideas or meaning by an IRS is typically achieved via a document representation scheme. The basic element fundamental to the representation of textual material inside an IRS has been the "keyword". As far as the human brain is concerned it is unrealistic to treat a "keyword" as the sole representative of a concept. Though detail of how exactly the brain formulates ideas (or concepts) and how they are structured in the brain for efficient storage and reasoning are not clear, its remarkable accuracy and robustness to deal with imprecision and vagueness of the IR problem suggests that they would possibly be much more complex than simple keywords. Therefore in this work, we assumed that the formalism and the structure of an idea or a concept in the brain is more complex that it requires a more elaborate entity to represent it within an IRS and also that the concepts in the brain are kept in a more complex structure that retains the underlying interconnections between concepts. Furthermore this interconnected structure is assumed to be the key factor that allows the brain to disambiguate meaning of an individual concept with respect to the collective meaning of the context in which the concept being used and that helps its reasoning mechanism.

The framework provided by FCA [3,10] is the closest formalism we found that gets along with this line of thinking. 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 hierarchically ordered conceptual structure (i.e. a concept lattice structure). A concept lattice represents and conveys a broader picture of the knowledge that the combination of the individual concepts of the context possesses.

The adaptivity or the learning aspect of the human brain, with respect to IR, works in the following way. A document that is found to be not useful for a query in the past is unlikely to be tried again for the same or similar information need(s) in the near future (by the same individual). This is because the brain remembers at least the main concepts of a recently seen document, if not all of it, and that it knows that the document does not help the information need at hand. What this essentially means is that the experience gained at the early search sessions help later sessions. However, the brain does not retain all its memories forever. The memory fades away (forgets) over time. This forgetting feature, though looks undesirable, is an extremely useful property for the adaptation and also to prevent information explosion in the brain. We used a reinforcement learning strategy to mimic these properties in our IR model. Based on the relevance feedback information given by the user for the retrieved documents, significance of (formal) concepts and keywords that contributed for the retrieval of documents that the user finds useful are made stronger (increased) and the significances of the concepts and keywords of the rest of the documents (the ones that the user finds not useful or not interested in) are made weaker (decreased, as they have contributed towards a false hit).

In the following we first describe briefly how concept matching between a query and a document lattice is performed. The reader is referred to [17,18] for detail of document representation in Concept lattices, implementation of concept lattices in BAM structures and further detail of concept matching. Our reinforcement learning strategy is presented next followed by the experimental results of the system obtained on Cranfield collection and the conclusions.

2 Use of Concept Lattices in IR in the Past Compared to our Approach

Use of concept lattices in IR in the past has been mainly on developing browsing mechanisms for domain specific IR [1,4-9,12,14,15]. Typically, a single large concept lattice is created based on the keywords present in documents. Objects in concepts are the document identities (Identification numbers and attributes are the keywords in them. This formalism is not much different to the keyword based document categorization approaches, except for the organisation of groups of documents hierarchically in Concept Lattices. The user is provided with a starting node, and allowed to navigate the lattice by expanding the nodes and traversing between nodes. The stating node can either be the root node or any other node decided based on an initial keyword based search.

We have identified a number of disadvantages in the past approaches as found in the papers [1,4-9,11-16]. (1) the formulation of concepts as objects being document identities and attributes being the keywords present in those documents is rather unrealistic in terms of the way the human brain formulate, perceive and communicate concepts. (2) use of a single large concept lattice to represent the entire document collection is computationally very expensive, and as a result such systems are limited to smaller document collections. (3) most of the past models are limited for browsing only (4) Creation of the lattice needs complex lattice building algorithms. Also traversing in the lattice is expensive and increases proportional to the size of the lattice. (5) once created, the lattice is fixed and no learning facilities is provided in any of the past FCA based approaches. Our approach is different to those approaches at least in the following four ways:

1. concepts are formulated according to how the human brain might do it, i.e. by extracting subjects, topics or objects mentioned in the text as objects of the formal concepts and properties or attributes of them as attributes of the formal concepts. A set of ad-hoc rules (see in [17,18]) was developed to extract such features from natural language text based on syntactic structures of the sentences.
2. each document (query) is represented by a single (separate) lattice. Advantage of this is two fold; firstly it allows us to operate on smaller lattices rather than on a single large lattice, and secondly, it allows maintaining different weights for the same concept in different document representations.
3. concept lattices are encoded in BAM structures [13]. Learning a BAM with a concept lattice[2,17,18] is much more efficient than using complex lattice building algorithms. Also updating lattice representations of documents with additional concepts is much easier with BAMs [18] and no node traversing overhead involved.
4. finally, we have employed a reinforcement learning strategy based on relevance feedback information to interactively learn document representations by (1) learning significances of concepts (formal concepts and keywords) and (2) updating the lattices with additional (query) concepts that would help retrieving documents.

3 Concept Matching Between a Query and a Document

A node in a Concept Lattice represents a formal concept of the type described above. The process of reasoning the usefulness of a given document to an information need (query) is achieved based on the concepts common to the query and the document as similar to the way common features/terms are used in conventional IR, i.e. based on how similar the nodes in query concept lattice to the nodes in the document concept lattice (i.e. node matching). Following example illustrates node matching between a query and a document.

Consider the context of planets given in Table 1. This context can be regarded as a representation of a document that talks about the solar system. The concept lattice of this context is given in Fig. 1 (Right). Consider an information need containing the object "Mars" and a "moon" as a property of it. Assume that the concept lattice of this query contains a node representing the formal concept *{Ma}→{my}* (see Fig. 1. (left). Note that we disregard the rest of the nodes in the query lattice in this illustration. Document nodes that match with this query node are also shown in Fig. 1.

Planet	Size			Distance from Sun		Moon	
	small	medium	large	near	far	yes	no
Mercury	x			x			x
Venus	x			x			x
Earth	x			x		x	
Mars	x			x		x	
Jupiter			x		x	x	
Saturn			x		x	x	
Uranus		x			x	x	
Pluto		x			x	x	
Neptune	x				x	x	

Table 1. A Context of Planets

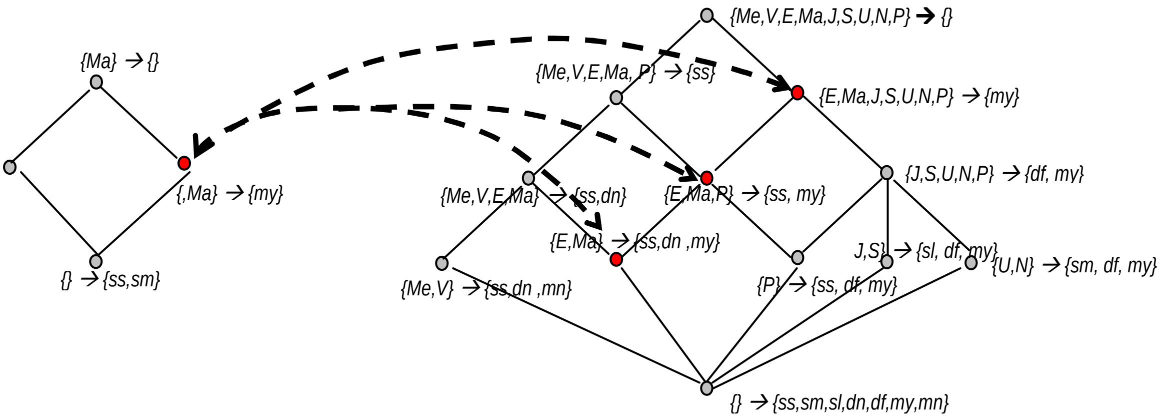


Fig. 1. Node Matching between two Concept Lattices

As can be seen in the above figure (Fig.1), comparing query nodes with document nodes is not as simple and straightforward as comparing just simple terms or keywords. Here, we need to maintain the consistency of our treatment to certain terms as objects and certain others as attributes. In addition, we wish to take into account the superiority/generality of concepts within the concept hierarchy in order to match more specific concepts and also to avoid duplications. However, problems of natural language such as synonymy, polesemy, and other problems related to the variability of vocabulary which cause mismatches between concepts and the size variability (number of objects and attributes) between query and document concepts etc. causes 100% match between a query and a document node impossible. Instead a mechanism to perform partial matching is required.

3.1 Partial Matching

We define a partial match between two concepts as a concept *m* consisting of objects common to the query and the document extents in its extent and attributes common to the query and the document intents in its intent. This can be formally defined as given below:

Let $q=<A,B>$ and $d=<C,D>$ be two formal concepts, then the partial match between the two concepts is given by the concept $m = <A \cap C, B \cap D>$, where A,B,C,D are sets of terms of which terms in *A* and *C* are interpreted as objects and terms in *B* and *D* are interpreted as attributes according to the FCA formalism.

3.2 Concept Weighting

Partial matching of concepts needs a mechanism for determining the significance of a partial match. Assigning significances weights to full concepts doesn't help it. Therefore weights were assigned to single object-attribute pairs (unit-concepts) as shown in Fig. 2.

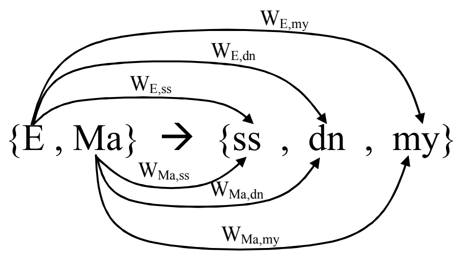
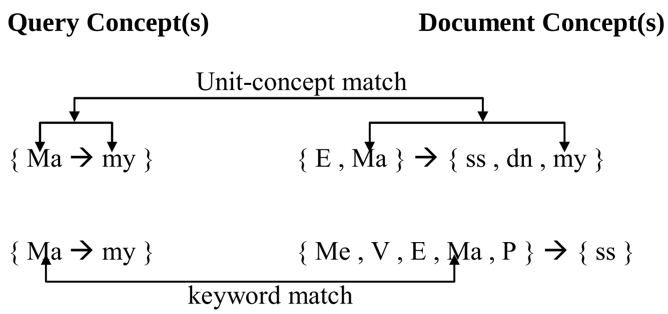


Fig. 2. Concept Weighting

The object-attribute pairs (unit concepts) in *m* determine how similar the two concepts *q* and *d* are. If the query concept is identical to the document concept (ideal case) a complete node match occurs, i.e. $m=q=d=<A,B>=<C,D>$.

4 Keyword Matching

It is not uncommon for two concepts to share a common object or attribute but not both (hence no unit-concept match occurs). This could happen due to various reasons such as the problems caused by term/element mismatches or the two nodes actually represent two distinct concepts though the same term/element has happened to be used by chance. Whatever the reason, we do not want to ignore the possible contributions that such common features (keywords) between two nodes may make in retrieving a relevant document. Such a single object or attribute matches are considered as "keyword" matches and significance weights are maintained for individual keywords as well. The following diagram illustrates the difference between concept and keyword matches between concepts (nodes).



Note that the keyword match of "Ma" will eventually be pruned out (in this case) as unit-concept matches with "Ma" in them do take place in this example.

5 Learning Strategy

Our learning strategy works by accepting the user feedback in the form of *yes* and *no* (i.e. accepts a document as relevant or reject it as irrelevant) and accordingly improving the document representation.

Traditionally, relevance feedback has been used to reformulate the query with additional information to support IR. Even though it has shown as much as 20% improvement on recall and precision, one of the drawbacks of this approach is that it does not support learning. Important user decisions (user feedback) obtained are used only within one query session for searching one information need. The results gained by relevance feedback at one query session are usually not available for the subsequent query sessions, because the IR system does not retain them or their implications. A separate learning mechanism is required to make such systems adaptive.

Instead, in our model user feedback is used to update the document representations and the modifications made to the documents are retained. We expect the document representations to converge to a well representative set of concepts (for each document) over a period of time. Such a set of concepts, indeed, will become more personalized to the vocabulary and the writing style of the end user, as (i) it is the concepts of the user formulated queries that are amended into relevant document representations and (ii) it is the user's relevance assessments that are used for reinforcing significance weights of init-concepts and keywords in documents.

Our reinforcement learning process works as follows; if user says a particular (retrieved) document is relevant to a given query, all unit-concepts of the query that are not present in the document are added to the document representation with an initial weight value. In case that a particular unit concept of the query is already present in the document, we consider it as an important unit concept (because it has made some contribution for the document's retrieval in the first place) and therefore its weight is increased by a small amount (∇w) as described in section 5.1 below. The concept addition may result in unnecessary unit concepts getting into the document's representation. But we expect such unnecessary concepts to be penalized by our learning strategy and end up with low weights in the long run.

Conversely, if a user says a particular (retrieved) document is not relevant to the query then the weights of matching units (unit-concepts and keywords) that are common to the query-document pair (i.e. those that contributed for the document's retrieval) are decreased to say that those units, though present in both query and the document, are not very important to decide the relevancy of the document to the query.

5.1 Significance Weights and Step Size of Weight Changes

Both keyword and unit-concept weights are initialized at the beginning with the initial value of 2.5 and are allowed to learn over the user interactions. The range of a weight was selected arbitrarily and constrained to positive values between [0.1-5.0]. The minimum value of the weight range was set to 0.1 instead of zero (0) to avoid complete ignorance of the existence of a feature.

The step size is determined proportional to the current value of the weight. The idea is to make the changes based on how far it is from the boundary towards the direction of change (i.e. to the to boundary if rewarding or to the bottom boundary if penalizing. See Fig.3 below.). This makes learning faster if the difference between the current weight and the boundary towards which the modification is made is larger and slower otherwise. The weight modification formula is:

$$W^{new} = W^{old} + \eta \Delta W$$

Where,
$\Delta W = W^{max} - W^{old}$ if a positive reinforcement or
$\Delta W = W^{old} - W^{min}$ if a negative reinforcement
and η is the learning rate.

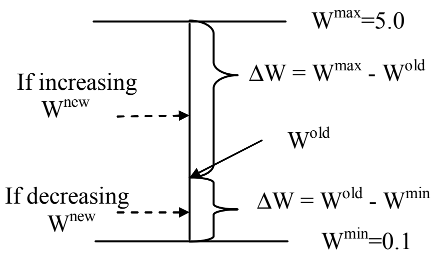


Fig. 3. Weight Modification Steps

5.2 Learning Rates for Rewarding and Penalizing

The learning rate is a constant that tells the proportion of the weight difference to take into account for the actual step size of the modification. The nature of IR is such that, only a few of the many documents retrieved are judged as useful by the user. Therefore only those few documents that the user decides useful for his information need are rewarded. All the other documents in the retrieved set are regarded as false hits and therefore are penalized. As a result, on average, the weights of concepts/keywords tend to be negatively reinforced more times than they are positively reinforced. This imbalance of negative and positive reinforcements may lead all weights to end-up with the minimum weight value (0.1) allowed, if not dealt appropriately. A way to get round this problem is to use different learning rates for positive and negative reinforcements. Deciding precise values for positive (η) and negative (β) learning rates is difficult as it depends on a number of factors including the number of queries, composition of the queries and user judgments etc. Based on the results of a few preliminary experimentations on Cranfield collection, they were set to take η=0.04 and β=η/3=0.0133.

Fig. 5 illustrates the learning strategy described above using an example. In that the query consists of only two unit-concepts and the documents retrieved for this query contains two relevant documents (Doc35 and Doc50) and two not relevant documents (Doc20 and Doc100). Matching units, weight updating and concept addition (according to our learning mechanism) are shown in the diagram for the documents Doc35 and Doc20.

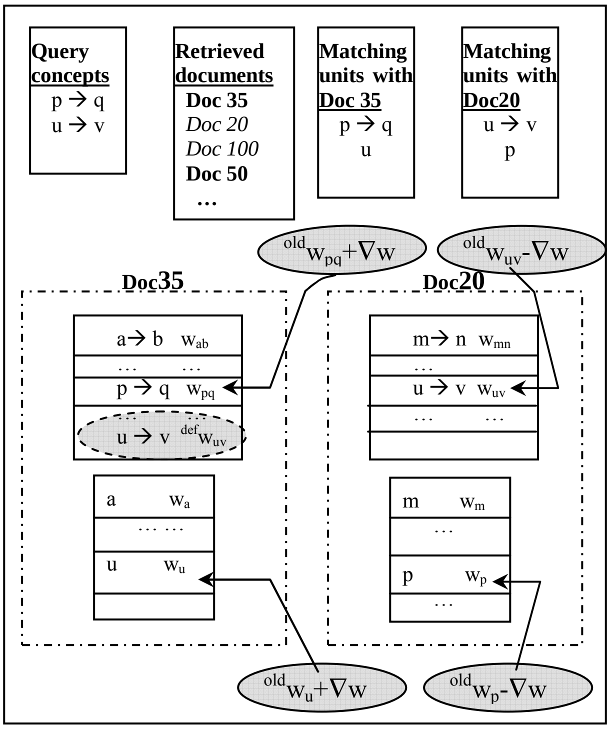


Fig. 5. Reinforcement Learning Strategy

5.4 Informative Factors of Comparison Units

The weight reinforcement strategy described above treats all units (unit-concepts and keywords) as equals, i.e. the weight of each unit-concept/keyword is reinforced by the amount decided based on the learning rate and the current value of the weight as described above regardless of their informative-ness. Since, not every concept/keyword is equally informative, we made use of 4 levels of informativeness to take into account the informativeness of a comparison unit based on the number of terms that they possess. The weights of concepts/keywords are re-weighted using 4 pre-decided weighting factors (we call them "Informative Factors") at the time of computing similarity measure (RSV). Note that they are not used (directly) in weight modifications. Given below are the four different levels of informative-ness we considered. They are listed in the increasing order of their informative-ness and the values experimentally chosen (not optimized) for informative factors are given within brackets.

a. Single-term keywords (1.0)
b. Key Phrases (Keywords with more than one term) (1.6)
c. Unit-concepts with single- term components (both object and attribute) (2.0)
d. Unit-concepts with multi-word components (at least one component constitutes more than one term) (3.0)

6 Retrieval Process and Similarity (RSV) Computation

Retrieval process begins when a user issues a query (a natural language expression). This query expression is pre-processed for concept extraction and a concept lattice of the extracted concepts is then set-up. Then the concept lattice of each document in the collection (one at a time) is also set-up and the nodes of query concept lattice are compared with the nodes of the document concept lattice for partial matching.

6.1 Candidate Node/Concept Pairs for Comparison

Not all query concepts match with all document concepts and therefore attempting to perform such a matching is not worthwhile. Instead, we extract "candidate" concept pairs to match between the query and the document based on the presence of common unit-concepts and keywords between them. The candidate concept extraction process works mainly by looking for the most specific concept in the document lattice for each query object (i.e. using object concepts). Attribute concepts (i.e. the most generic concept containing a given attribute) are also used in the cases where a related object concept is not available in the document. During this process, we make sure to extract the most specific concepts wherever possible and also not to extract the same concept pair more than once. Also we avoid extracting document (query) concepts that are general (in the general-specific hierarchy in the concept lattice) to any of the already extracted document (query) concepts to match with the same query (document) concept. In addition, in case if an object or attribute in the query appears as both object and an attribute in a document representation, we check whether there is any order relation (in the concept hierarchy) between them in order to avoid matching two related document concepts with the same query concept. Only the most specific concept is considered for matching in such cases.

However there are some cases where we find the same query object appearing both as an object and attribute in document representations, but they represent two different ideas/concepts (i.e. they are not related in the concept hierarchy). In this case, the attribute concept given by document lattice for the quey object is also taken in to account as a candidate concept to matched with the object concept obtained from the query lattice for the query object, in addition to the object concept obtained from the document lattice.

6.2 Similarity Measure (RSV)

Candidate concept/node pairs decided (as described above) to match between a query and a document are detected are extracted from the corresponding lattices and compared for partial concept matching (i.e for unit-concept matching) and for keyword matching (in the absence of a unit-concept match with a given common keyword). Matching unit-concept pairs and keywords are then subject to pruning for removing duplicates. The sum of the significant weights of remaining unit-concepts and keywords (after pruning) is taken as the similarity measure (RSV) between the query and the document.

7 The Evaluation/Test Strategy

Given the unavailability of an appropriate evaluation methodology for evaluating the dynamic properties of interactive IR systems, we were compelled to use our own test strategy (which we call incremental Learning-Testing Strategy) for testing the performance dynamics of the system as it learns and gains experience. This was achieved by splitting the set of queries into two sets (training and testing sets) and then training the system on a (cumulative) subset of training queries at a time (i.e at each training session). Splitting the query set into training and testing tests so that they both equally represent query space (in terms of desired properties such as degree of overlap in relevance assessments and degree of similarity) is a difficult task. This was done based on the degree of overlap in relevant assessments, as it is the most important factor that helps interactive learning in our model. Degree of overlap was measured in terms of number documents assessed as relevant to each query. Out of the 225 queries available in the Cranfield collection 65 queries were used for testing and 160 queries for training. More queries were allocated to the training set simply because we needed more queries to create more training-testing (sub) sessions. This gives a fairly representative set of queries for testing in terms of cross-relations, but does not guarantee that queries are equally distributed in terms of their similarity and expressiveness in natural language (length).

A training set for each training phase was created by adding 40 randomly selected queries from the set of full training set (of 160 queries) into the training set used at the previous training session. No query is picked more than once. So, the numbers of queries trained at the four training sessions were 40, 80, 120 and 160. Each query was iterated 20 times at each training session. The order of presentation of queries to the system was made random. At the end of each training session, the system was tested with the 65 test queries and the similarity measures were recorded for each query-document pair.

8 Results

Fig. 6 shows that the performance of the system (i.e. non-interpolated Average Precisions) increases considerably over training. Note that the shape of the curve varies depending on the amount of learning taken place at each training session and how much those learning helps retrieving relevant documents for test queries, but the starting and ending points remain the same. This is because the selection of training queries for sub training sessions from the full training set (160) is done randomly.

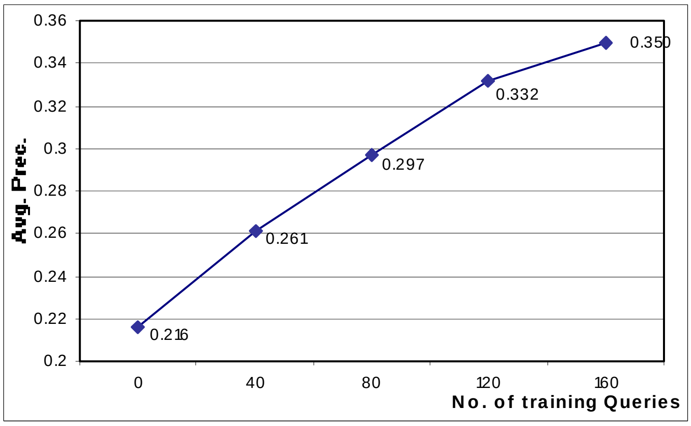


Fig. 6. Average Precision Over Training

Fig. 7 shows the P-R curves of test results obtained after each training session. It provides more evidence to the performance gains shown above by the system over training.

The performance gain shown by the system is a combined result of the use of unit-concepts & keywords and the use of our reinforcement learning strategy. Contribution of individual components (of matching and learning) towards this result is further analysed and compared below in figures 8,9 and 10.

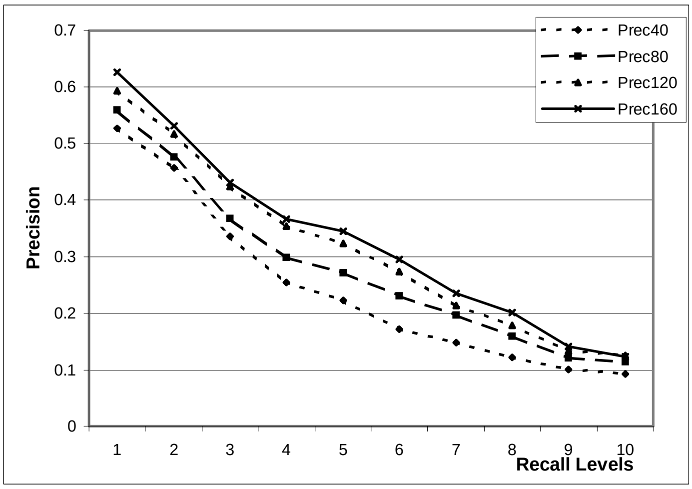


Fig. 7. P-R Curves at Different Levels of Training

Fig. 8 shows that the concept matching alone without any learning (the flat curve) is not of much use. Problems with concept extraction and with mismatches caused by the vocabulary differences and word ambiguity in natural language are the main causes for the poor performance of concept matching (only). These problems are severe in our case compared to simple keyword matching, because (i) Concept extraction from source documents is more complex and difficult, as it needs identification of a two terms or phrases one as an object and the other as a property possessed by the object (ii) Mismatch problem is doubled in our case because a concept match needs both the object and attribute constituents of a query concept (unit-concept) to match with an equivalent in a document.

Allowing the system to learn concept weights and also adding query concepts to relevant documents each alone have shown some improvements of performance over learning but not sufficient enough (Fig. 7). Concept learning does not (and it is not expected to) solve the two main causes of the poor performance stated above. Though, concept addition helps documents to learn (through user interactions) different ways users might refer to them (experience) and there by helps both word mismatch problem and poor concept extraction, it has not shown a significant improvement. This is mainly due to the fact that these results were based on testing unseen queries and lack of sufficient overlaps in the collection (i.e. use of same unit-concepts to represent similar documents) has hampered evaluation of the main property our learning strategy - retrieval of a document by a query as a result of the document being reinforced (updated) by another query. However the interesting point here is that they both show increasing trends in performances. As a result of these positive improvements of each component, their combination has shown a significantly better improvement.

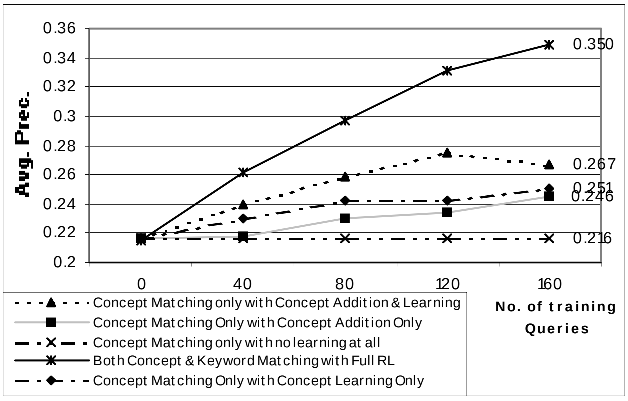


Fig. 8. Contribution of Learning on Performance

Interestingly the performance of the system has increased when only the keyword-learning component is used despite the fact that Keyword matches at each testing session were the same in this case. It is the same set of queries that were tested on the same collection and no concepts (and hence no keywords) are added to the documents during training. This is solely a result of keyword weight learning only. Our learning strategy seems to have assigned higher weights to the keywords that were significant at least in terms of the document distinguishing power.

Finally, the performance curve of the system in its full capacity shows that taking keyword matching into account helps improve performance. The reasons for this improvement are that (1) keywords help initial picking up documents for reinforcing and (2) keyword matches (that takes place in the absence of unit-concept matches) help increasing the similarity scores (RSVs) of documents and thus help ranking document with more features common with the query above the ones with less features. The second point is valid only if more keyword matches occur with relevant documents than with non-relevant documents. This is a well-known observation first made by B. Croft. Though no experiments were targeted at examining the validity of this feature, the improvements showed by the system with keyword matching evident its validity. In case if more keyword matches occurred with non-relevant documents, those non-relevant documents are pushed up in the rank list and as a result the performance of the system should have degraded.

Fig.9 and Fig.10 compare the performance of the system with its different learning components when only concept matching and keyword matching (respectively) are considered.

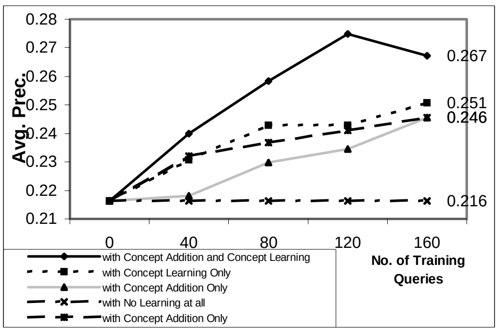


Fig. 9. Concept Matching Only

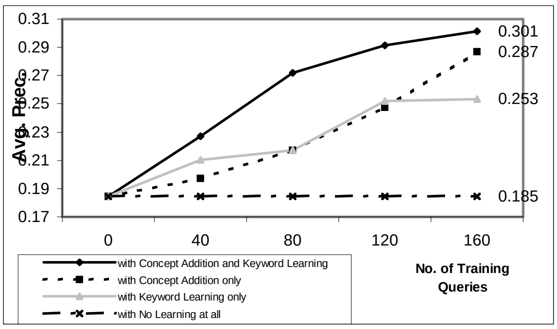


Fig. 10. Keyword Matching Only

They both show that combining all three learning components give better performance results for both concept matching (only) and keyword matching (only). This result further confirms that the better performance shown by the system in its full capacity (in Fig. 1) is a combined result of all learning components on both concept matching and keyword matching but not a result of a subset of them.

However note that not all the keyword matches that takes place in the case of testing for "keyword matching only" becomes keyword matches when testing for "both keyword and concept matches" as the keywords that participate in concept matches are considered duplicates and pruned out. Therefore, the test results of “both keyword and concept matching” are not the same as the sum of the two cases of “concept matching only” and “keyword matching only”.

9 Conclusions

We have shown firstly a way of using more elaborate and true concepts for creating more meaningful representations of textual material and using them for explicit concept matching; secondly a radically different approach of using concept lattices in IR and its feasibility; thirdly the importance of an interactive learning strategy and the effectiveness of retaining the learnt knowledge for future use; and finally, the advantage of using a hybrid approach that takes into account both concept matching, keyword matching together with concept addition and weight learning for developing an IR system.

A main characteristic our system is that it becomes more and more tuned to its environment or strictly speaking to its inputs as it learns. Consistency of training examples i.e. the consistency of users in terms of the use of vocabulary in query formulation and making relevance assessments is essential for our system to converge or better tuned for its inputs. Consistency is maximized when only one user uses the system. Essentially, this makes the system customized to its only user thus making it more personalized. On the other hand learning in multi-user environments help learning more exhaustive and better-generalized representations. In particular it helps the system to learn different possible ways of formulating queries (i.e different ways of referring into the same document) by different end users with different vocabularies. However the consistency among the users in making relevance assessments is essential for convergence. In an environment with more inconsistent users, the system dynamics and therefore retrieval performance may vary rapidly in time. As a result a given user may not be guaranteed the same relevant document for the same query issued at a later attempt. According to these observations, the system is likely to perform better in more personalized (single user) environments as well as in multi-user environments with similar or consistent users. Indeed it has the potential to outperform conventional keyword based systems in such environments.

Finally, this research is concluded with the following comment/recommendation. This work was a first step towards making use of more elegant concepts as much similar as possible to the formulation of concepts in the human brain and also allowing end users to (implicitly) decide the significances of concepts in the documents through an interactive learning strategy. The difficulty of automatic concept extraction from text and lack of sufficient background information in documents for building more complete concept hierarchies can be mentioned as the major drawbacks that crippled our investigation of the full potential of concept matching within the framework of FCA. However, despite these drawbacks, the performance results obtained are impressive and encouraging. In particular this work shows a way of performing true concept matching and the feasibility of using FCA in a different and more advantageous way to that of existing FCA based approaches. We are optimistic of the potential of the FCA framework to deliver better performance provided a better, more meaningful document/query representations are created through the incorporation of background knowledge using external knowledge sources and extraction of more meaningful concepts from text using future advancements of NLP technology. Such work is left for the future.

References

1. Becker, P., Hereth, J., Stumme, G.: ToscanaJ - An Open Source Tool for Qualitative Data Analysis. In: Workshop proceedings of FCA KDD workshop of the 15th European Conference on Artificial Intelligence (ECAI’02), July 21-26 2002, Lyon, France (2002)
2. Bělohlávek, R.: Representation of Concept Lattices by Bidirectional Associative Memories. Neural Computation Vol 12 No.10 October (2000) 2279-2290
3. Burmeister, P.: Formal Concept Analysis with ConImp: Introduction to the basic features.Technical Report, Technische Hochschule Darmstadt, Germany (1998)
4. Carpineto, C., and Romano, G.: A Lattice Conceptual Clustering System and Its Application for Browsing Retrieval. Machine Learning , 24(2) (1996) 95-122
5. Carpineto, C., and Romano, G.: Order-theoretical ranking. JASIS Vol.51 No.7 (2000) 587-601
6. Cole, R., Eklund, P.W.: Scalability of Formal Concept Analysis. Computational Intelligence, Vol 2, No. 5 (1993)

7. Cole, R.J., Eklund, P.K.: Application of Formal Concept Analysis to Information Retrieval using a Hierarchically Structured Thesaurus. International Conference on Conceptual Graphs, ICCS'96, Sydney (1996) 1-12
8. Cole,R., Eklund,P., Stumme,G.: CEM-A Program for Visualization and Discovery in Email. In: D.A. Zighed, J. Komorowski, J. Zytkow (Eds): Proceedings of PKDD 2000, LNAI 1910 (2000)
9. Cole,R., Eklund,P., Stumme,G.: Document retrieval for Email Search and Discovery using Formal Concept Analysis. Journal of Applied Artificial Intelligence, Vol. 17, No. 3 (2003)
10. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer-Verlag Berlin Heidelberg , ISBN 3-540-627771-5 (1999)
11. Godin, R., Missaoui, R., April, A.: Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. International Journal of Man-Machine Studies, 38 (1993) 747-767
12. Kim, M., Compton, P.: A Web-based Browsing Mechanism Based on Conceptual Structures. 9th International Conference on Conceptual Structures (ICCS 2001), Guy W. Mineau (Eds.), California USA, CEUR-WS, Stanford University (2001) 47-60
13. Kosko, B. : Bidirectional Associative Memory. IEEE Transactions of Systems, Man and Cybernetics, 18(1) (1988) 49-60
14. Lindig, C.: Concept-Based Component Retrieval. Working Notes of the IJCAI-95 Workshop: Formal Approaches to the Reuse of Plans, Proofs (1995)
15. Priss, U.: A Graphical Interface for Document Retrieval Based on Formal Concept Analysis. In: Proceedings of 8th Midwest Artificial Intelligence and Cognitive Science Conference, AAAI Technical Report CF-97-01 (1997) 66-70
16. Priss, U.: Lattice-based Information Retrieval. Knowledge Organization, Vol.27, 3 (2000) 132-142
17. Rajapakse, R.K., Denham, M.: A Concept based Adaptive Information Retrieval Model using FCA-BAM combination for concept representation. In: Proceedings of the 24th BCS-IRSG European Colloquium on IR Research (ECIR'02), March 25-27, 2002, Glasgow, UK (2002)
18. Rajapakse, R.K., Denham, M.: Information Retrieval Model Using Concept Lattices for Content Representation. In: Workshop proceedings of FCA KDD workshop of the 15th European Conference on Artificial Intelligence (ECAI'02), July 21-26 2002, Lyon, France (2002)

This paper appeared in the proceedings of the 7th World Multiconference on Systemics Cybernetics and Informatics (SCI'2003), Orlando, Florida USA, July 2003. Also appeared in the workshop proceedings of the "User Modeling, Information Retrieval and Machine Learning" workshop of the 9th International Conference on User Modeling (UM'2003).