

Information Retrieval Model using Concept Lattices for Content Representation

Rajapakse, R.K¹ and Denham, M¹

Abstract. This paper reports our investigation into the applicability of concept lattices, that are based on the theory of formal concept analysis and lattice theory to the representation of concepts/ideas expressed in natural language (English) text, and the encoding of such concept lattices in Bidirectional Associative Memories (BAMs) in order to effectively and efficiently manipulate concepts in documents to support Information Retrieval. Features or concepts (formulated as defined in FCA) of each document (and query) are represented in a separate concept lattice and are weighted separately with respect to the documents allowing the same concept to have different weights in different documents. 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 its retrieval. A reinforcement learning strategy based on relevance feedback information makes the similarity of relevant documents stronger and nonrelevant documents weaker for each query.

1 INTRODUCTION

Almost all existing IR models make use of either single terms (keywords) or phrases (two or more adjoining terms) to represent the basic unit of matching: a *concept* [17]. Exceptional to these are the network models that learn the implicit relations between keywords to build a form of implicit concepts to assist the IR task and the Knowledge based techniques in which additional (related) keywords are incorporated with the keywords extracted from the text to enhance the query or document(s).

In our work, great deal of effort was extended to the direct extraction of meaningful ideas/concepts from text and representation of them in a formal framework to assist the IR task. We investigated the applicability of concept lattices that are based on the theory of formal concept analysis and lattice theory to the representation of concepts/ideas expressed in natural language (English) text. An interesting feature of concept lattices is that the concepts are hierarchically organized according to a subsumption order relation defining a specificity-generality relationship between concepts. We use this property to match most specific concepts between queries and documents.

In our implementation, concept lattices are encoded in Bidirectional Associative Memories (BAMs) in order to effectively and efficiently manipulate concepts to support Information Retrieval. The BAM has been recognized as a simple neural network that is able to learn a concept lattice in its two-layered architecture. Once trained, it can be used to extract the most

specific or most generic concept from the concept structure for a given set of interested objects or attributes by presenting them into the corresponding layer of the BAM.

We view the document retrieval process as a continuous conversation between queries and documents. Through these conversations, documents are allowed to learn a consistent set of significant concepts resulting for queries similar to already seen ones to retrieve the documents that were retrieved for those similar queries and judged as relevant in the past (improves recall) and not to retrieve the documents that were retrieved for those similar queries but judged as not relevant (improves precision). The learning strategy used to achieve this goal is one based on relevance feedback by which the document representation is improved (on the fly) to make the similarity between a query and a retrieved relevant document stronger and that between a query and a retrieved nonrelevant document weaker.

In the following, first we present the theoretical background for the techniques used (FCA & BAM) and discuss the suitability of the proposed representation scheme for Information Retrieval. The analogy between the representation of formal concepts in a structured manner in a concept lattice and the representation of ideas/concepts in human brain in the process of human understanding is briefed. The IR processes, including what is (what concepts) matched with what (concepts) in the document, how a similarity value is computed and how the relevance feedback is used for reinforcement learning are detailed at the later part of the paper.

The work reported here is an ongoing research work. A prototype of the model has been implemented and tried on Cranfield collection. Preliminary results are encouraging. Future work includes incorporating a query reformulation mechanism, further improvements to the concept extraction process and finally evaluation of the performance of the model against published results.

2 FORMAL CONCEPT ANALYSIS (FCA)

Formal Concept Analysis was proposed by Rudolf Wille in 1982 [11,19] as a mathematical framework for performing data analysis. It structures data into units that are formal abstractions of “*concepts*” of human thought, allowing meaningful and comprehensible interpretation. FCA models the world as being composed of *objects* and *attributes*. It is assumed that an incident relation connects objects to attributes. The choice of what is an object and what is an attribute is dependent on the domain in which FCA is applied. Information about a domain is captured in a

¹ Centre for Neural and Adaptive Systems, University of Plymouth, UK

“formal context” - a formalization that encodes only a small portion of what is usually referred to as a “context”. FCA models the specificity-generality relationships between two related concepts by means of an order relation. The formal definitions of FCA are given below.

Definition 1: A Formal Context $K = (G, M, I)$ consists of two sets G (set of objects) and M (set of Attributes) and a relation I between G and M . (See Table 1)

Definition 2: A Formal Concept is defined on a Formal Context as a pair of sets (A, B) , where A is a set of objects and B is a set of attributes, in which attributes in A are maximally possessed by the set of objects in B and the sets of objects in A is the maximal set of objects possessing the set of attributes in B . Formally, a pair of sets (A, B) where $A \subseteq G$ and $B \subseteq M$ is a formal concept if: $A^\perp = B$ and $B^\perp = A$ (Completeness Constraint).

Where $A^\perp = \{ m \in M \mid gIm \text{ for all } g \in A \}$ (i.e. the set of attributes common to all the objects in A) AND $B^\perp = \{ g \in G \mid gIm \text{ for all } m \in B \}$ (i.e. the set of objects which have all attributes in B). gIm means “ g is related to m ” (e.g. a binary relation).

The set of all concepts of a context (G, M, I) is denoted by $B(G, M, I)$. This consists of all pairs (A, B) where $A \subseteq G$ and $B \subseteq M$ s.t. $A = B^\perp$ and $B = A^\perp$.

Definition 3: Specificity-generality order relationship :If (A_1, B_1) and (A_2, B_2) are concepts of a context, then (A_1, B_1) is called a sub concept of (A_2, B_2) , if $A_1 \subseteq A_2$ (or $B_1 \supseteq B_2$). This sub-super concept relation is written as $(A_1, B_1) \leq (A_2, B_2)$. In other words, a subconcept is a concept with less objects than any of its superconcepts. Equivalently, a subconcept is a concept with more attributes than any of its superconcepts.

2.1 Concept Lattice

A lattice is an ordered set V with an order relation in which for any given two elements “ x ” and “ y ”, the supremum and the infimum elements always exist in V . Furthermore, such a lattice is called a *complete lattice* if supremum and infimum elements exist for any subset X of V [11,19]. The fundamental theorem of FCA states that the set of formal concepts of a formal context forms a complete lattice [9,11,19]. A complete lattice of *formal concepts* is called a *concept lattice*. Figure 1 shows the concept lattice of the context given in Table 1.

Table 1. A Context of the Planets

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	

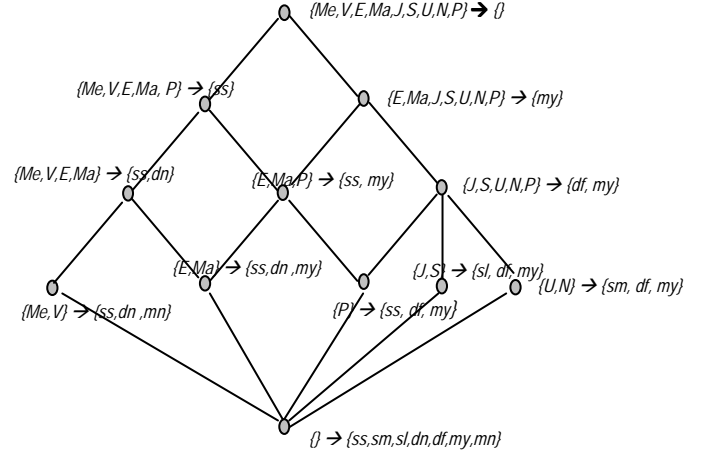


Figure 1. Concept Lattice of the context in Table 1

2.1.1 Join/Meet Concepts

A Concept Lattice can be visualized as a graph with nodes and edges (links) (Figure 1). Concepts at the nodes from which two or more lines run up are called *meet* concepts (i.e. nodes with more than one parent) and concepts at the nodes from which two or more lines run down are called *join* concepts (i.e. nodes with more than one child) (See Figure 1).

A Join concept groups objects sharing the same attributes and a meet concept separates out objects that have combined attributes from different parents (groups of objects). Each of these join and meet concepts creates a new sub- or super- category (class) of a concept.

3 BIDIRECTIONAL ASSOCIATIVE MEMORIES (BAMS)

Based on the early associative memory models [1,14], Kosko [15,16] proposed a bi-directional associative neural network called Bidirectional Associative Memory (BAM). A BAM consists of two layers of neurones. The states (activities) of the neurones in the two layers are denoted by x_i ($i=1, \dots, k$) and y_j ($j=1, \dots, l$) respectively, where k and l are the number of neurones in the two layers. The states x_i and y_j can be encoded either a binary (0 or 1) or a bipolar (+1 or -1) encoding. Each (i^{th}) neuron of the first layer is connected to each (j^{th}) neuron of the second layer by a connection weight (see Figure 2). A real threshold θ_i^x (θ_j^y) is assigned to the i^{th} neuron of the first layer (j^{th} neuron of the second layer) respectively.

A number of different weight computation schemes have been suggested for setting up the connection weights in a BAM (training or learning a BAM). Amongst are the one proposed by Kosko[15], the originator of *BAMs* and that proposed by Radim Bělohlávek [2]. The aim of learning is to set the parameters (weights) of the network so that a prescribed training set of patterns is related in some way to the set of all stable points. Kosko’s learning strategy to determine weights w_{ij} from a training set $T = \{ \langle X_p, Y_p \rangle \mid p \in P \}$ is $w_{ij} = \sum_{p \in P} \text{bip}(x_i^p) \cdot \text{bip}(y_j^p)$. Thresholds of all units are set to zeros [15]. Where, $\text{bip}()$ maps 1 to 1 and 0 to -1 changing the binary encoding to a bipolar. $P = \{1, 2, 3, \dots, \text{no. of training patterns in } T\}$.

3.1 Dynamics of BAMs

Given a pair $\langle X, Y \rangle = \langle \langle x_1, \dots, x_k \rangle, \langle y_1, \dots, y_l \rangle \rangle \in \{0,1\}^k \times \{0,1\}^l$ of patterns of signals, the signal X is fed to the first layer to obtain a new pair $\langle X, Y' \rangle$, then Y' to the second layer to obtain $\langle X', Y' \rangle$, and so on. The dynamics is given by the formulas:

$$y'_i = \begin{cases} 1 & \text{for } \sum_{j=1}^k w_{ij} x_j > \theta_j^y \\ y_j & \text{for } \sum_{j=1}^k w_{ij} x_j = \theta_j^y \\ 0 & \text{for } \sum_{j=1}^k w_{ij} x_j < \theta_j^y \end{cases} \quad x'_i = \begin{cases} 1 & \text{for } \sum_{j=1}^l w_{ij} y'_j > \theta_i^x \\ x_i & \text{for } \sum_{j=1}^l w_{ij} y'_j = \theta_i^x \\ 0 & \text{for } \sum_{j=1}^l w_{ij} y'_j < \theta_i^x \end{cases}$$

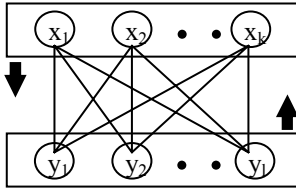


Figure 2. Structure of a BAM

The pair of patterns $\langle X, Y \rangle$ is called a stable point if the states of neurones, when set to $\langle X, Y \rangle$, do not change under the above defined dynamics. Using appropriate energy function, Kosko [16] proved that such a network is stable for any weights w_{ij} and any thresholds θ_i^x, θ_j^y . Stability means that given any initial pattern $\langle X, Y \rangle$ of signals, the network eventually stops after a finite number of steps (after feeding signal from layer to layer back and forth).

3.2 BAMs for Storing Concept Lattices

In general, there are BAM stable points that cannot be interpreted as formal concepts [2]. However, using the weight computation given below, Radim Bělohávek showed that BAMs can learn concept lattices [2]. He proved that "there is a BAM corresponding to each concept lattice $B(G, M, I)$ such that the set of all concepts of $B(G, M, I)$ is precisely the set of all the stable points of the BAM. The weight computation used for the proof is given by:

$$w_{ij} = \begin{cases} 1 & \text{if } \langle g_i, m_j \rangle \in I \\ -q & \text{if } \langle g_i, m_j \rangle \notin I \end{cases} \quad \text{for } i=1, \dots, k, j=1, \dots, l$$

where $q = \max\{k, l\} + 1$. All the thresholds are set to $-1/2$.

We use this weighting scheme in our implementations to train BAMs with concept lattices.

3.3 A Training Set for a BAM to Learn a Concept Lattice

A training set T consists of a set of concepts in the form (A, B) , where elements of A comes from the set of objects G and elements of B from the set of attributes M of the context (G, M, I) . Here the set G contains all the objects necessary to define the extents (A) of the concepts (A, B) in the training set and M contains all the attributes necessary to define the intents in the training set. This means that the elements in T are defined on (G, M, I) . A conceptually consistent training set however needs its training patterns to obey the fundamental rule (completeness constraint) of a formal concept: $A^1=B$ & $B^1=A$ which ensures that the set of formal concepts in the training set are storable in the BAM.

Any given set of arbitrary concepts (a Training set T) which satisfies the above condition defines a concept lattice on the context (G, M, I) formed by all the objects and attributes of the concepts in T . This training set is a subset of $B(G, M, I)$ (set of all the concepts of the context (G, M, I)). It also is a subset of the concept lattice $\underline{B}(G, M, I)$ of the context (G, M, I) .

Described above is what a training set in general should be. However, Radim's weighting scheme does not need constructing training sets as per above description. Instead, It only needs to know all the individual objects in the context and all of the attributes possessed by each of them. Given these information, we can simply set the weights between object nodes and attribute nodes to $+1$ if the object possesses the attribute or $-q$ if it does not possess the attribute.

3.4 A BAM in Operation - an Example

The following two figures illustrate three things. Firstly, Figure 3 illustrates what the nodes represent and how the weights are set between object and attribute nodes, Secondly, Figure 3 and Figure 4 together shows how the BAM works in its forward pass and backward pass. Note that, the nodes in gray are the active nodes, dashed-lines represent links with negative ($-q$) weights and solid lines represent links with weight $+1$. Lastly, figure3 & figure 4 together demonstrates the important feature of a BAM in which for a given input pattern (in this case a pattern with only *Mass* active) the BAM returns the most specific concept available in the Lattice containing the given input object(s); in this case $\{E, Ma\} \rightarrow \{ss, dn, my\}$.

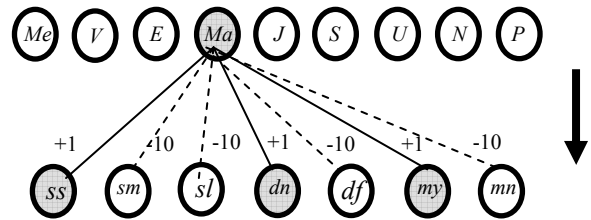


Figure 3. Forward Pass

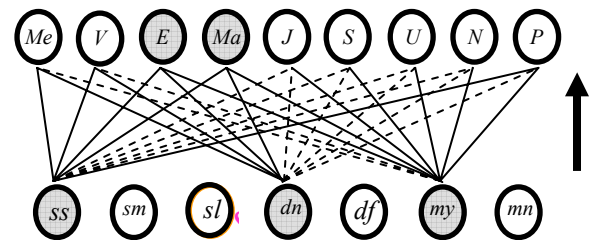


Figure 4. Backward pass

3.5 What Actually a BAM Learns and Returns?

Given a set of training patterns consisting of each individual object in the context and all of its attributes (note we use Radim's method), a BAM learns the underlying concept lattice. Join and meet concepts that are inferred by the patterns in the training set are automatically detected and learnt by the BAM. For example, given the nine training patterns, one for each planet (with all of

their corresponding attributes), the concept lattice given in figure 1 is derived.

Unlike purely feed-forward neural network architectures, BAMs can accept input patterns from either end (layer). We can present a pattern with objects to the first layer (referred to as the object layer) or a pattern with attributes to the second layer (referred to as the attribute layer). The BAM returns the most specific concept containing all the objects of the input pattern in the first case (Figure 3 & Figure 4) and conversely, the most generic concept containing all the attributes of the input pattern in the second case. We use this interesting property for extracting candidate concept pairs from query and document BAMs to match between (as described in section 6.3).

4 TOWARDS CONCEPTS/IDEAS EXPRESSED IN NATURAL LANGUAGE

A number of questions have to be addressed before employing FCA into IR tasks. These include: what is an object; what is an attribute; what is a concept/idea; how can we extract objects, attributes and concepts from textual material; what is the analogy between the order relationship defined for formal concepts and the ideas/concepts extracted from textual documents; what is the role of join/meet concepts in human understanding of natural language text etc. This section attempts to answer these questions.

4.1 Abstracting Ideas/Concepts in Human Understanding

The theory of concept lattices has been founded [11,19] based on a traditional understanding of concepts by which a concept is determined by its extent and intent. The extent of a concept (e.g. *DOG*) is the collection of all objects covered by the concept (the collection of all *dogs*), while the intent is the collection of all attributes (e.g. *to bark*, *to be a mammal*) covered by the concept. This interpretation of a concept can be directly employed in representing concepts expressed in natural language. The formation of an idea or a concept in the human mind during the understanding of natural language text is initially triggered by the objects (physical or conceptual) and attributes (properties of objects) in the text followed by the overall context of the subject being read and the reader's background knowledge of the subject.

4.1.1 *The Two Entities: Objects and Attributes*

In general, an object corresponds to a subject, and attributes modify the meaning of the object to express the context in which the object is being used. For instance, a particular set of attributes of the object *DOG* may support the context of say *eating patterns of dogs*, while certain other set of attributes may support say the *sleeping patterns of dogs*. FCA captures these two important aspects/features, the *subject* and the *context* in its two entities *objects* and *attributes* to formulate an idea/concept. This makes FCA a good candidate for formulating and manipulating "human thoughts" (concepts) within computer systems.

4.2 Similarity of super-sub order relationship in formal concepts and natural ideas/concepts

A sub-concept in FCA, defined as a concept with less objects and more attributes than its superconcept(s) means we need more

attributes to define something specific compared to the less attributes needed to define something generic. Ideas/concepts stored in human mind may be structured in a similar manner, whereby we need more detail to learn a more specific idea/concept. For instance to define a *bird* we need to say it *can fly* and it *has a beak*, in addition to the information necessary to say that its an *animal*. However the frequently used generic attributes (in this case the attributes to specify that a bird is a living animal) are not usually used to express an idea/concept during normal human conversations and writing. They are implicit. An Average human brain has gained the necessary background knowledge in understanding frequently used common ideas when expressed just by the (name of) the main object(s) of the subject. For instance, we never try to define what an animal is during conversations, instead we simply use the term "*animal*". At some point during our learning process (implicit or explicit), we have absorbed all the necessary attributes to understand what an animal is. It is obvious however that encoding concepts/ideas in a computer requires all the information necessary to distinctly identify a particular idea/concept to be explicitly specified. These background general ideas/concepts are analogous to the superconcepts and the specific sub-categories/concepts of them are analogous to the subconcepts defined in the FCA formalization.

4.3 The role of Join & Meet concepts in Human understanding

Categorizing common objects (or ideas) together is a natural phenomenon in human understanding. If you are asked to name few animals you can give a vast number of different animals as examples for animals. If you are then asked to name few animals who are carnivorous, you certainly have no problem of naming a set of animals who eat meat. You may have given names of some of these carnivorous animals as examples for animals for the first question as well. This means your brain knows how to categorize the same set of objects depending on the context (depending on the attributes that each object possesses). It is the same phenomenon that the *meet* and *join* concepts model in FCA.

These similarities between the formalization of concepts in FCA and ideas/concepts of human understanding/thought process mentioned above suggest that the way the humans formulate concepts, structure them and use them in the process of understanding and expressing ideas is analogous to the way concepts are formulated in FCM and are structured in a Concept Lattice.

5 FEATURE EXTRACTION FOR CONCEPT GENERATION

Conventional indexing schemes largely ignore the inter index term relationships. Capturing such information leads to the formation of a structure of ideas/concepts in the text representing the conceptual content of the text better than individual index terms do [10]. This task partly belongs to the more complex "natural language understanding" problem. However, the fact that a deep and complete understanding of the text may not be mandatory for IR [10] makes it sufficient to work with a shallow and partial representation of document contents that is capable of correctly classifying documents as relevant or irrelevant for a given query.

Thought individual noun-phrase analysis has proven to be superior to full text indexing [10], that alone is not sufficient

enough in our case as we are interested in relationships between noun groups as well. We make use of techniques such as part-of-speech (POS) tagging and noun phrase analysis together with a set of selected propositions that frequently appear between noun groups as connectors in order to extract a well representative set of concepts (objects and attributes) from text.

Our approach begins by first assigning identification labels to each Noun and Verb group and thereby creating a string with these labels to represent the syntactic structure of each sentence. Text that does not belong to a noun or verb group is left unlabelled (and used as it is). For example, the syntactic structure of the sentence “[The dog] (chased) [the cat] away” would be: $NG_1|VB_1|NG_2|away$, where NG_1 is the label for the first noun group (in this case [The dog]), NG_2 is the label for the second noun group ([the cat] and VB_1 is the label for the first verb group (chased). “|” is the separator character. Note that noun groups are indicated by square brackets and verb groups by curly brackets.

Syntactic structure of each noun/verb group is also extracted for each group in terms of POS tags.

E.g. 1. $DT|NN$ is the syntactic structure of the noun group [The Dog]
 2. $DT|JJ|NNP|NNP$ is the syntactic structure of the noun group [The famed London Bridge]

where DT, JJ, NN, NNP are the POS tags for a determinant, an adjective, a noun word and a proper noun respectively (from Penn Treebank tag-set).

We then use a number of rules for identifying objects and their attributes based on the syntactic structures of the sentences and their components. These rules were developed after examining the syntactic information of the sentence structures and their components (noun groups and verb groups) on a considerably representative set of documents. They are described under the following three subtopics.

5.1.1 Prepositions (connectors) Between Two Chunks

Use of noun phrases for document indexing in general do not capture the relationships between noun groups. We capture such relationships between those noun groups that are connected by a prepositional connector. The connectors considered are : *in, on, of, with, to, into* and *from*. The relationships inferred by these connectors between two noun groups can be interpreted as *object-attribute* relationships. The set of connectors/prepositions mentioned and the rules that decide the roles (as objects or attributes) were also selected after analysing experimental results on the syntactic structures of sentences. The set of connectors and associated rules are not complete by any means, but consist of the most frequent and useful connectors and rules complying to the common writing styles.

Usage of connectors is such that the roles of the terms (as objects and attributes) in most connectors (such as “in” and “at”) usually appear in a particular written order (object-connector-attribute OR attribute-connector-object). However there are connectors (such as “of”) in which object (attribute) can appear on either side. E.g. in [millions] of [visitors] visitors is more suitable as the object and millions as the attribute, but in [crowd] of [1000], crowd is more suitable as the object and 1000 as a property of the crowd.

Most noun groups are comprised of several adjectives, adverbs and noun words etc. (E.g. *The_DT red_JJ old_JJ car_NN*). It may be more meaningful to use the noun words of two related noun groups in creating a concept(s) rather than taking entire group(s). The following example(s) shows how this can be done.

e.g. 1. [*the_DT red_JJ car_NN*] of [*the_DT tall_JJ man_NN*]
 2. [*a_DT lobby_NN*] of [*dark_JJ marble_NN walls_NNS*]

In the first example, creating a concept as $\{tall\ man\} \rightarrow \{red\ car\}$ causes matching problems if the same adjectives (*red* and *tall*) are not used by the user in his query statement. A better concept would be: $\{man\} \rightarrow \{car\}$. Note that determinants (like *A, The*) are always ignored and we use the convention $\langle extent \rangle \rightarrow \langle intent \rangle$ to write a concept in which the extent (and the intent) can contain more than one object (attribute) written comma separated.

Processing syntactic structures of noun groups described below (section 5.1.2) extracts $\{man\} \rightarrow \{tall\}$ and $\{car\} \rightarrow \{red\}$ as concepts. The first two concepts will then be joined during concept formation resulting the single concept $\{man\} \rightarrow \{car, tall\}$. This allows partial matching with query concepts $\{man\} \rightarrow \{car\}$ or $\{man\} \rightarrow \{tall\}$. However using this way alone also causes a problem if a particular document talks about more than one distinct car (say in different colours) that should be identified separately. The above mentioned method constructs a concept with the object *car* together with all colours as its attributes. Therefore it is useful to use both ways, i.e. in case of this example $\{tall\ man\} \rightarrow \{red\ car\}$, $\{man\} \rightarrow \{car, tall\}$ and $\{car\} \rightarrow \{red\}$. In which case, if the document talks about a *green car*, a concept will be formed taking *green car* as a single attribute of an object with which it relates. During concept matching, we must be careful enough to take into account only the most expressive matching concept(s) in case if concepts sharing the same terms (in their extents or intents) match between a query and a document, i.e. if both $\{tall\ man\} \rightarrow \{red\ car\}$ & $\{man\} \rightarrow \{car\}$ matches with a query, we should take only $\{tall\ man\} \rightarrow \{red\ car\}$ into account as otherwise we may be duplicating the same idea/concept twice.

5.1.2 Syntactic Structure of Noun Groups

As noted above, most noun groups contain several noun words and adjectives/modifiers within them. They are further analysed using POS tags attached to each word in order to detect useful object-attribute relationships between the words within them.

For example, if the syntactic structure of a noun group is $DT|JJ|NN$ or $JJ|NN$ (e.g. “*The_DT fat_JJ man_NN*”) then a concept is formed as $NN \rightarrow JJ$ (i.e. $\{man\} \rightarrow \{fat\}$). Here “*fat*” is an attribute of the object “*man*”. This is read as the object “*man*” has the property “*fat*”.

In case of noun groups comprising more than one noun words, concepts are formed according to the following rule.

$$[noun1\ noun2\ noun3] \Rightarrow \begin{matrix} noun3 \rightarrow noun1, noun2 ; \\ noun3 \rightarrow noun1+noun2 \text{ and} \\ noun2+noun3 \rightarrow noun1 \end{matrix}$$

5.1.3 Possessive relationships

The possessive relationships between two noun words indicated by a trailing ‘s or s’ are also detected and processed separately to form concepts containing all the words in the noun group up to the quote (‘) as the object and the words after the tag POS as the attribute.

- E.g.: 1. [The_DT man_NN's_POS hair_NN] => {Man} → {hair}
 2. [Seattle_NNP's_POS central_JJ business_NN district_NN]
 => {Seattle} → {central business district}

An interesting outcome of these concept extraction rules is that the same concept happen to be formed by certain different ways of writing the same idea (verbal groups). For instance, both noun groups [The_DT man_NN's_POS hair_NN] and [The_DT Hair_NN] of_IN [the_DT man_NN] lead to the extraction of the same concept {man} → {hair}.

The above-mentioned methods/rules, though not perfect, have shown to extract a reasonably well representative set of concepts from text. Additionally, they happen to deal with certain verbal groups (as described above) as well - a desired feature.

6 USING THE PROPOSED REPRESENTATION SCHEME FOR IR

An IR system needs to compute a similarity measure or retrieval status value (RSV) for each query-document pair to represent the similarity or appropriateness of the document to the query. Typically, RSV values are computed using the importance of matching entities (keywords/concepts). This makes it necessary to identify the basic units/entities to match between a query and a document and model their importance.

In general concepts of the kind we use consist of more than one object (in its extent) and more than one attribute (in its intent). (i.e. more than one unit concepts) We cannot expect such detailed concepts to match exactly between queries and documents due to various differences between them. A way to get-round these matching problems is to match single object-attribute pairs (unit concepts). The more unit concept matches between two query and document concepts, more similar the two concepts are (in IR terminology the similarity between the two concepts are stronger.)

6.1 Unit(s) of comparison/matching

We have recognized the following two units/entities as important to be used as the basic units for matching.

1. Unit concepts (a single object-attribute pair)
2. Keywords/keyphrases (any term/phrase that appears as an object or an attribute in unit concepts)

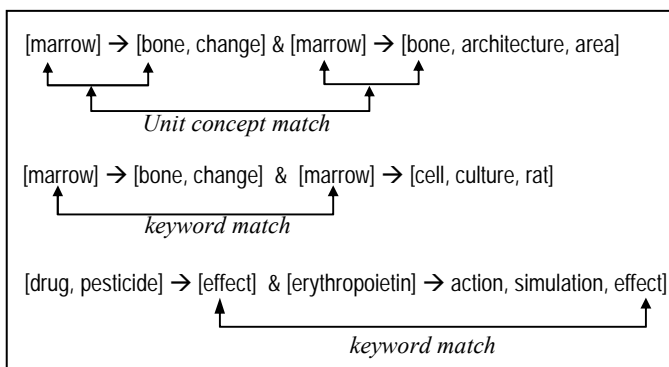


Figure.5. Possible unit matches between concepts

The reason for distinguishing the above two is because we are more interested in matching object-attribute pairs (i.e. theoretically speaking, an object and the context in which it is used) than just

single keywords or keyphrases. However, keywords are also important to pick up relevant documents (to a given query) in the absence of matching unit concepts due to various reasons including the use of different vocabulary by the user (to formulate the query) and inadequate detail in the query (short queries). (See figure 5).

6.2 Importance of Features

A given document generally consists of more than one concept (of the kind we extract from text). Some of these may be general concepts that might not be very important to distinctly identify the document while some others may be much more important. It is customary to model the importance of concepts and keywords by means of weights. More important concepts/keywords may be assigned bigger weights compared to the less important ones.

The weights of unit concepts model the importance of the object-attribute pairs within a particular document while the weights of the keywords/key-phrases model the importance of keywords/key-phrases in the document. It is important to maintain them with respect to each individual document, as a concept that is important to identify and retrieve one document may not be important to identify another document. So the presence of the said concept in the second document should not miss-recognise it as relevant for a query containing that concept. Therefore weights should be maintained separately for each individual document allowing the same object-attribute pair (unit concept) or the same keyword/key-phrase to have different importance weights in different documents.

6.2.1 Weight Computation

The traditional approach for weight computation is to use term/concept frequency statistics. In this case, a term/concept stated more number of times in a document is considered more important than the ones less stated (term frequency). An alternative (machine learning) approach is to initialise the weights at the beginning with a pre-decided value and then allow them to learn. This requires the IR system to have a weight learning/updating mechanism.

The first method, in general does not directly support an adaptive IR system as the importance of matching units are static. Instead we opted for the second method in our model. We use a reinforcement learning strategy based on the relevance information (user feedback) for this purpose (section 7).

6.2.2 Importance of the types of matching units

Not every matching unit is equally informative. We have identified some differences in the degree of informative-ness of both types of the comparison units (unit concepts & keywords). Therefore, weights of matching units are further weighted to take into account the differences of informative-ness. Following are the 4 different levels of informative-ness we have identified in our comparison units. They are listed in the increasing order of their informative-ness.

- Single-word keyword
- Keyword (phrase with multiple words)
- Unit concepts with single-word components (objects and attributes)
- Unit concepts with multi-word components

6.3 Candidate Concepts/Nodes to Compare

After having represented both a query and a document in concept lattices, the next step is to decide what concepts/nodes should be compared for similarity computation. Not all the concepts in a given pair of a query and a document lattices match each other. Therefore attempting to compare each and every concept in the query lattice with each and every concept in the document lattice is not worth the effort. In addition such a matching strategy fails to make use of the important knowledge of the order relationship structure, which has already been captured and encoded in the document/query representations (in BAMs). This is where the importance of using BAMs matters. Instead of using conventional traversing and searching techniques, we make use of the BAM's properties to directly access the desired concepts in the concept hierarchy by presenting appropriate inputs to the appropriate layer.

We are interested in the two properties of the BAM: (1) given an input pattern (with some objects), the BAM gives the most specific concept in the Lattice that contains all the objects of the input in its extent; and (2) converse of the above in which given an input pattern (with some attributes), the BAM gives the most generic concept in the Lattice that contains all the attributes of the input in its intent. The concepts given by the BAM at those two cases when the input pattern contains only one object (or one attribute) are called *object concept* of the input object and *attribute concept* of the input attribute respectively.

We have developed an algorithm (not detailed here) to extract candidate concepts to match between queries and documents based on these ideas. It ensures to extract the most specific concepts (for a given input) wherever possible and also do not extract the same concept pair more than once. Also it avoids 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. It also takes into account the concept pairs that lead to keyword matching.

6.4 Similarity Calculation

Similarity between a query and a document is computed by taking into account of all the matching units (unit concepts & keywords/keyphrases) between all the candidate query-document concept pairs compared. Sum of the weights of matching units (unit concepts and keywords) is taken as the measure of the similarity between the query and the document. Note that, in case of two units with one included in the other or one is a more expressive form of the other (e.g. presence of a matching keyword in a matching unit concept), we always ignore the less informative one take into account the most informative unit to avoid possible duplication.

6.5 Thresholding

The RSV values computed for each query-document pair is then subject to thresholding in order to decide which documents should be presented to the user as the retrieved set of documents. We use a kind of dynamic thresholding strategy by taking into account the total number of unit concepts available in all the candidate query concepts considered for comparing with document concepts (i.e. the total number of unit concepts we are looking for in the candidate document concepts) to compensate the varied sizes (size of a query is determined by the number of unit concepts in its

representation) of queries. This value is multiplied by a predefined base threshold value. Use of a base threshold value allows us to experiment on the best thresholding value to be used by varying the base threshold.

$$(\text{Base Threshold}) \times \left(\text{No. of unit concepts in all the candidate query concepts considered for matching between a given query-document pair} \right)$$

7 CONCEPT LEARNING

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) 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 learn them. A separate learning mechanism is required to make such systems adaptive (to learn).

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 personalised to the vocabulary and the writing style of the end user, as it is the concepts of the user formulated queries that are amended into relevant document representations. Our reinforcement learning process works as follows:

If user says a particular (retrieved) document is relevant to a given query, all the unit concepts of the query are amended to the document representation. 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 pre-decided amount (\sqrt{w}). Unit query concepts not present in the document are simply added to the document representation with an initial weight value. This may result in unnecessary unit concepts getting into the document's representation, but we expect such unnecessary concepts to end up with low weights in the long run. Note that, BAMs are updated to highlight the modifications in order to combine the new unit concept(s) with appropriate concepts and place them at the appropriate positions in the concept hierarchy.

Conversely, if a user says a particular (retrieved) document is not relevant to the query then we examine for matching units (unit concepts and keywords) that are common to the query-document pair (i.e. those matching unit concepts and keywords that contributed for the document's retrieval) and their weights are decreased (by a pre-decided value) 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.

The following figure (figure 6) 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.

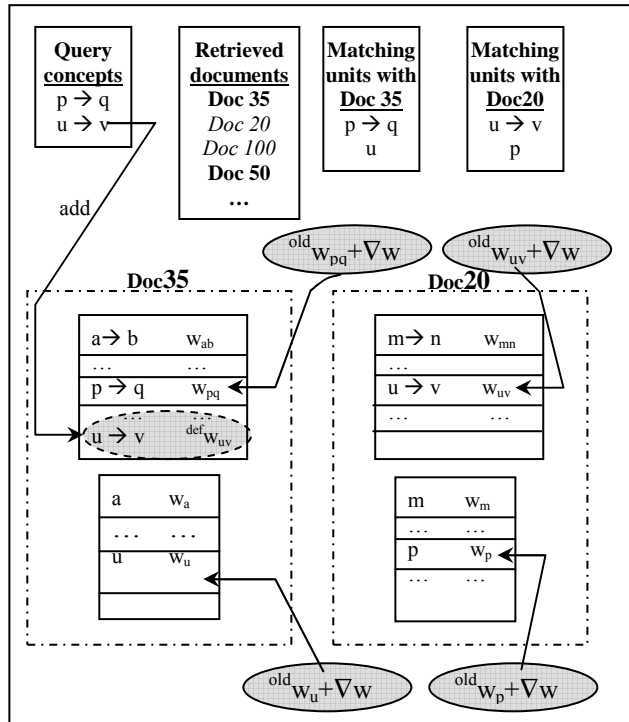


Figure 6. Learning Strategy

8 PRELIMINARY RESULTS

Preliminary results of the implementation of our prototype model were promising for well descriptive queries and documents. We have tried our prototype implementation (only) on Cranfield collection (1400 documents and 225 queries). We find it hard to compare the performance results of our model against published results of other well established IR models due to the representation and learning differences. (eg. our IR model cannot operate on traditional representation schemes and traditional IR models cannot operate on our representation scheme).

Nevertheless, a set of results published by Carpineto [5] is given below (in Table 2) to show how the figures of comparison metrics look like. This particular set of published results was interesting to us as it includes results of Carpineto's concept lattice based IR model [5]. Though his model is completely different to that of ours, it is one of the very few models that use concept lattices for IR which can operate on free text, domain independent documents. His experiments have been on the CASM and CISI collections while ours on Cranfield collection - another difference that makes our results incomparable to Carpineto's. Nevertheless, our results are shown against his results to show that they follow the same order.

Table 2. Published Results

	CASM			CISI			CRAN
	BMR	HCR	CLR	BMR	HCR	CLR	Our Model
Average	0.320	0.231	0.253	0.164	0.127	0.162	0.365
Precision at 5 points	0.346	0.342	0.412	0.269	0.280	0.337	0.379
Precision at 10 points	0.304	0.298	0.240	0.266	0.254	0.286	0.288
Precision at 20 points	0.238	0.202	0.164	0.239	0.209	0.234	0.184

9 RELATED WORK

Concept lattices have been used for information retrieval by a number of researchers including Godin et al., [12,13]; Carpineto and Romano, [3,4,5]; Cole and Eklund [6,7]; Cole et al [8]. Most of these researchers have employed FCA to support user interface design, to help the user navigate through the concept lattice to locate desired documents. Almost all of them formulate as having a set of documents (as the objects) in the extent and their keywords as attributes in the intent, and represent the entire document collection in a single large concept lattice. Using a single concept lattice to represent the entire document collection demands high memory requirements and computational power for traversing and maintaining the Lattice. In contrast, we attempt to capture concepts as an analogue of how the human cognitive brain process might work and use individual much smaller concept lattices for each document. The most related work to that of ours is the Lattice-based data structure proposed by Merwe and Kourie [18]. Their data structure is very much similar to our representation scheme in the sense that we both embed concept lattices in two layered data structures. However, Merwe's paper lacks a retrieval process to compare with ours.

10 DISCUSSION

We expect that in the long run, when used on a given collection of documents with a good representative set of natural language queries, the system should stabilize on a better set of representative concepts for each document in the collection, through which a better recall and precision can be achieved. We expect our system to perform better than the traditional models in terms of effectiveness, given representative query and document concept lattices.

One of the major advantages of the proposed model is that it does explicit concept matching, a desired feature, via its unit concept matching. This we suppose is in line with how humans read a document and match their information needs with the contents of documents to decide whether they are useful.

The second advantage of the model is the use of important user feedback (relevance feedback) for concept learning. Unlike the

traditional approach that uses relevance feedback for query reformulation, we use relevance feedback for improving document representations. Those improvements learnt from past experience are retained. We use a reinforcement learning strategy that allows

the documents to learn concepts on the fly improving both recall and precision. Upgrading weights of matching units in the retrieved relevant documents and degrading weights of matching units in the retrieved but not relevant documents make the concepts that support retrieval of relevant documents to become stronger and concepts that causes not-relevant documents to retrieve to become weaker. In addition, concepts that are not present in retrieved relevant documents are added into the document representation allowing the document to be retrieved with a higher precision next time for the same query and also allowing a second query to pick

the document as a result of concepts added by the first query being matched with the second query. The addition of query concepts into documents will indeed add unnecessary concepts (say rubbish) into documents as one would argue, but the idea is that those rubbish will end up with low weights in the long run making not much impact on a document's retrieval. An additional feature of the learning mechanism is that it leads towards a more personalised system as you go due to the fact that it is the user who decides the importance of the weights of concepts through relevance feedback.

Another useful feature of the model is its decentralised document representation. Since we represent documents as individual concept lattices based on the local information available in the documents, they are completely independent of each other. This kind of a representation will be appealing for decentralised document collections in which documents can reside in several computers.

Moreover, our model is expected to operate on less terms/words (compared to full text tf-idf models) and is able to operate in a domain independent manner. One can plug-in domain specific knowledge bases to support initial concept extraction to improve document representation in view of obtaining better the performance, but what is interesting here is that even without such external sources of domain knowledge, the model should work moderately well improving its performance as it learn.

Drawbacks of the model include the expensive initial document representation (concept extraction) process, expensive candidate concepts/nodes extraction process to match between queries and documents, and lack of global knowledge about the document collection. Expensiveness of feature extraction is a trade-off between how much informative the concepts you need to extract from text and how much efficient the system should be. The candidate concept/node extraction to match between is severe only if the document lattices are too big. On average they are not big in the collection we used. However, a pruning process to remove rubbish (insignificant concepts with low weights) from documents will help to maintain document lattices within a manageable size.

We have no information (at present) to comment on the storage requirements over a long period of time as the document representations tend to grow over the time. But, we expect the system to converge to a fixed document representation. As mentioned above, keeping track of the less-used or never-used unit concepts, and thus perceived unimportant concepts in document representations and removing such less useful concepts from the documents will help making the model efficient in terms of retrieval speed, and also making the system to operate on low storage capacities.

An important future enhancement to the model for better performance will be to incorporate a query enhancement/reformulation mechanism. At present, the model does not have a query reformulation component to enhance the initial user queries with additional related concepts/keywords. In addition, an improved concept extraction process is desired.

REFERENCES

[1] Amari, S. (1972). Learning patterns and pattern sequences by self-organizing nets of thresholding elements. *IEEE Trans. On Computers*, 21(11), 461-482.
 [2] Belohlávek, R (2000): Representation of Concept Lattices by Bidirectional Associative Memories, *Neural Computation* Vol 12 N.10 October 2000. Pp 2279-2290

[3] Carpineto, C., & Romano, G. (1996): A Lattice Conceptual Clustering System and its Application to Browsing Retrieval. *Machine Learning*, 24, 1-28.
 [4] Carpineto, C., & Romano, G. (1996): Information retrieval through hybrid navigation of lattice representations. *Int. Journal of Human-Computer Studies*, 45, 553-578.
 [5] Carpineto, C. & Romano, G. (2000): Order-theoretical ranking, *JASIS* vol51 No. 7 587-601 (2000).
 [6] Cole, R and Eklund, P.W. (1993): Scalability of Formal Concept Analysis, *Computational Intelligence*, Vol 2, No. 5, 1993. <http://www.int.gu.edu.au/kvo/papers/>
 [7] Cole, R.J. and Eklund, P.K. (1996): Application of Formal Concept Analysis to Information Retrieval using a Hierarchically Structured Thesaurus, <http://www.int.gu.edu.au/kvo/papers/> International Conference on Conceptual Graphs, ICCS '96, Sydney, 1996, pp. 1-12, University of New South Wales, 1996.
 [8] Cole, R.J. Eklund, P.K. Stumme, G.: CEM-A Program for Visualization and Discovery in Email, <http://www.int.gu.edu.au/kvo/papers/> In D.A. Zighed, J. Komorowski, J. Zytkow (Eds), *Proc. of PKDD 2000, LNAI 1910*, pp. 367-374, Springer-Verlag, Berlin, 2000.
 [9] Darmstadt University of Technology: Formal Concept Analysis S/W, http://www.mathematik.tu-darmstadt.de/ags/ag1/Software/software_en.html
 [10] Evans, D.A. and Zhai, C. (1996): Noun-Phrase Analysis in Unrestricted Text for Information Retrieval, *Proc.s, 34th Annual Meeting of the Association for Computational Linguistics (ACL'96)*, pp. 17-24, Santa Cruz, CA.
 [11] Ganter, B. Wille, R. (1999) : Formal Concept Analysis : Mathematical Foundations, ISBN 3-540-62777-5 Springer-Verlag Berlin Heidelberg 1999
 [12] Godin, R., Gecsei, J., & Pichet, C. (1989): Design of a browsing interface for information retrieval. *Proceedings of the 12th Int. Conf. on Research and Development in IR (ACM SIGIR'89)*, pp.32-39. Cambridge, MA, ACM.
 [13] Godin, R., Missaoui, R., April, A. (1993): Experimental comparison of navigation in a Galois lattice with conventional information retrieval methods. *Int. Journal of Man-machine Studies*, 38, 747-767.
 [14] Hopfield, J.J.(1984) : Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Natl. Acad. Sci. U.S.A.*, 81, 3088-3092.
 [15] Kosko, B. (1987) : Adaptive bidirectional associative memory. *Applied Optics*, 26(23), 4947-4960.
 [16] Kosko, B. (1988) : Bidirectional associative memory. *IEEE Trans. Systems, Man and Cybernetics*, 18(1), 49-60.
 [17] Lewis, D. (1992): Feature Selection and Feature Extraction for Text Categorization, *Proceedings of Speech and Natural Language Workshop, 1992*
 [18] van der Merwe, F.J. & Kourie, D.G. (2001): A Lattice-Based Data Structure for Information Retrieval and Machine Learning, *ICCS'01 International workshop on Concept Lattices-based KDD*.
 [19] Wille, R. (1997): Conceptual Graphs and Formal Concept Analysis. In: Lukose, D. et. al. (eds.): *Conceptual Structures: Fulfilling Peirce's Dream*, *Proc. of the ICCS'97*. Springer, Berlin-New York (1997) 290-303 16

This paper was presented at the FCA KDD workshop of the 15th European Conference on Artificial Intelligence (ECAI'02) held in July 21-26 2002, Lyon, France, and appeared in the workshop proceedings.