Conceptual Clustering in Information Retrieval *

Sanjiv K. Bhatia
Department of Mathematics & Computer Science
University of Missouri – St. Louis
St. Louis, MO 63121-4499

Jitender S. Deogun
Department of Computer Science & Engineering
University of Nebraska - Lincoln
Lincoln, NE 68588-0115

Abstract

Clustering is used in information retrieval systems to enhance the efficiency and effectiveness of the retrieval process. Clustering is achieved by partitioning the documents in a collection into classes such that documents that are associated with each other are assigned to the same cluster. This association is generally determined by examining the index term representation of documents or by capturing user feedback on queries to the system. In cluster-oriented systems, the retrieval process can be enhanced by employing characterization of clusters. In this paper, we present the techniques to develop clusters and cluster characterizations by employing user viewpoint. The user viewpoint is elicited through a structured interview based on a knowledge acquisition technique, namely personal construct theory. It is demonstrated that the application of personal construct theory results in a cluster representation that can be used during query as well as to assign new documents to the appropriate clusters.

*This research is supported by the NSF grant IRI-8805875.

List of Tables

List of Figures

1 Introduction

Document clustering or classification deals with the physical and logical organization of textual items in a bibliographic collection. Clustering techniques are used in information retrieval systems to enhance the efficiency and effectiveness of the retrieval process [10]. In conceptual clustering, the objective is to identify classes of objects in a collection such that objects with the same set of features are assigned to the same class. In addition, each class must be properly characterized in terms of the features used for classification [17].

A system with a good clustering scheme can select a set of documents for further processing or ignore it as non-relevant to a query. Moreover, the documents that are stored physically close to each-other can be retrieved with a minimal disk-head movement. Therefore, document classification is essential in dealing with large collections and is an integral part of an information retrieval environment [25]. Ideally, the clusters reflect "natural association" between documents within a cluster [1]. Two documents are said to be *naturally associative* if they deal with similar subjects and therefore are collectively relevant or non-relevant to most queries. That is, if one document is relevant to a query, the documents that are naturally associative to the relevant document are also relevant to the same query. Documents in different clusters deal with relatively distinct subjects.

Automatic clustering of documents is achieved by identification of the subject addressed in the documents (keywords) on a syntactic level. Each document is compared to all other documents in the collection and a clustering is developed by aggregating those documents into a cluster that have a high degree of match. These techniques are based on strict match between document keywords and do not take into account query-based access patterns.

Yu, *et al*, [26, 27] hypothesized that with change in time, there is a change in a users' access patterns which should be captured in the clustering scheme. Their adaptive clustering scheme was termed *user-oriented clustering scheme* and is based on observing user response to queries. The response is used to assign those documents to the same cluster that are judged to be jointly relevant to the same query by the user. The technique was improved by Deogun and Raghavan [10, 19] to identify optimal clusters in which most of the documents are jointly relevant to a query.

User-oriented schemes attempt to classify a collection by establishing a consensus in the users' judgement of documents. This results not only in an increase in the initial setup time of the classification, but also causes problems in the maintenance, and requires extensive user cooperation to correctly classify each new document. Moreover, these systems do not provide for establishing the relationship between different classes. Furthermore, the user-oriented clustering schemes are based on passive observation of user responses to the results of a query. Therefore, the frequent queries have more effect on clustering compared to the queries that are performed rarely. The system leaves full responsibility of query formulation to the user. Since the documents are classified by observing query feedback, the documents that are not accessed often may not be classified appropriately. The inappropriate classification may pose a problem as the user access patterns change.

The process of clustering includes not only the initial organization of documents into appropriate clusters but also the development of a characterization for each cluster. The cluster characterization is meant to be a collective representation for the documents in the cluster, and is used in two important ways. First, it enables the system to select or ignore a cluster of documents for further processing in response to a query. Second, the system can assign a new document to an appropriate cluster by matching the document representation against cluster representations.

A clustering scheme should provide capability to assign new documents to the most appropriate cluster. To achieve this, the decision maker must be cognizant of different classes in the system and have the ability to relate them to the document being classified. A viable solution to the classification problem depends on the successful solution to three subproblems – creation of classes, determining the relationships between these classes, and the maintenance of the classification system [2].

We hypothesize that a user should not have to formulate queries to affect all documents in the collection. Instead, the user viewpoint is captured by knowledge acquisition techniques and then employed to cluster the documents. This ensures that the documents that are jointly relevant to a query will be assigned to the same cluster. Moreover, unlike the passive observation employed by the user-oriented systems [10, 18, 19, 26, 27], our system actively interviews the user to elicit his viewpoint. This viewpoint is used to develop a representation for each cluster. Individual documents are compared with cluster representations to determine the most appropriate cluster. This leads to a knowledge-based classification system.

In this paper, we develop a knowledge-based classification system for information retrieval using personal construct theory. The system is based on an automated interview process to elicit user viewpoint of some selected documents in the collection. The viewpoint is used to develop classes of concepts leading to document clusters. A characterization for each class that captures the user viewpoint is developed by mapping the problem of cluster characterization onto the problem of concept elicitation from experts in the development of expert systems. This is followed by development of rules that express and quantify the extent of match of an arbitrary document to the clusters. It is demonstrated that personal construct theory can be applied to solve the problem of assigning terms to cluster characterizations.

For the evaluation of clustering, we use the set of requirements for a good clustering scheme identified by Yu, *et al.*, [26]. According to their criteria, documents that are jointly accessed should be assigned to the same cluster. Moreover, highly correlated terms should belong to the same cluster. Second, clustering should not be based on the classification of queries, nor should an attempt be made to collect statistics on queries to be used for clustering. Third, the records or files should not be moved in response to the feedback on each query. Fourth, the clustering should be performed in polynomial time rather than exponential time as is the case with a number of classical clustering algorithms. Fifth, even though the individual queries are assumed to be independent, the clustering scheme should permit dependence between documents and keywords to allow the retrieval of a number of related documents in response to a query. Finally, the clustering scheme itself should be conceptually simple.

In Section 2, we describe the knowledge acquisition process using personal construct theory. The elicited knowledge is structured as a repertory grid that is analyzed to determine a rough classification (Section 3). The knowledge is refined in a post-interview session as presented in Section 4. The refined knowledge is correlated to system-observable features of documents to develop a classification in Section 5. In Section 6, we present the experiments for a step-by-step demonstration of our techniques which can be correlated with the earlier sections.

2 Knowledge Acquisition and Personal Construct Theory

The objective in document clustering is to aggregate those documents in a cluster that are retrieved in response to a query. The documents that are not relevant to similar queries should be assigned to different clusters. To achieve this objective, the system should know the queries that are similar to each other and also the documents that are retrieved in response to these queries. The queries can be in the form of concepts known to the user. The system can determine the similarity between queries if it knows the similarity between concepts. The similarity between concepts can be formalized as *clusters of concepts* by using knowledge acquisition techniques used in the development of expert systems. In this section, we describe the knowledge acquisition process based on personal construct theory.

Personal construct theory was pioneered by George Kelly [15] in modern clinical psychology. It is an elegant technique used by psychologists to gain insight into a person's behavior. According to personal construct theory, every action of a person is motivated by his environment and background. The environment consists of the properties of the objects surrounding the subject rather than the objects themselves. Therefore, the focus in personal construct theory is to determine the kind of environment from the subject's viewpoint that leads to an action. If the point of view of a person is known, his actions may be predicted with reasonable accuracy.

The use of systematic principles to elicit anticipatory knowledge makes personal construct theory suitable for any knowledge acquisition system to extract and structure subconscious knowledge of an expert for classification [20]. This was the motivation behind the successful adaptation of personal construct theory for knowledge elicitation by John Boose [5, 6]. He designed the Expertise Transfer System (ETS) to automatically interview an expert. ETS analyzes the elicited heuristic knowledge and constructs production rules (or actions) for incorporation in an expert system [6].

In a decision-making environment, the objects that have an influence on a person's actions, are known as *entities*. The significant properties of the entities, that are the cause of its influence on the behavior of a person, are called the *constructs*. The constructs are characteristics that can be rated on a linear scale and are exemplified by *friendly-unfriendly, good-bad*, and *clear-hazy*.

Constructs are elicited from a person by asking him to compare a few entities to identify a property that differentiates between those entities. An entity may influence the actions of a person to some degree with respect to a construct. This degree of influence can be quantified on a rating scale. If the rating is low, the entity does not possess the property named by the construct. Conversely, if the rating is high, the construct becomes important in the description of the entity. Two entities in a person's environment are considered to be similar if they have similar contributions towards all the constructs.

The knowledge is elicited from a person by using a *training set* of entities selected by the person. The knowledge elicitation process focuses on making the person enumerate the constructs through a structured interview using this training set. Since the user is interviewed to determine his opinion of the entities in the training set, it is important that he is well acquainted with all those entities. Also, the training set should contain at least one example of each construct considered important by the user. This is required to have an adequate case library of constructs. For this reason, the user is allowed to add entities and constructs at any stage during the interview.

The extent of relevance of constructs for different entities is assigned on a *rating scale*. The rating of all the entities on all the constructs results in a rectangular matrix known as a *repertory grid.*

2.1 Repertory Grid

The repertory grid provides a convenient framework for knowledge elicitation. It helps to explicitly document most of the factors influencing the expert's behavior with respect to entities. In the repertory grid, the constructs of a person are represented as a set of distinctions made with respect to a set of entities relevant to the domain.

In the knowledge acquisition interview, the user is asked to categorize a subset of the elements in the training set. This subset categorization is meant to force the user to bring forth and evaluate the difference between the elements in the subset. The system randomly selects at least three entities from the training set and presents them to the interviewee. The interviewee identifies an important concept such that two entities in this triad are jointly relevant or non-relevant to the concept. This concept forms one construct.

Next, the person is asked to assign a rating to all the entities in the training set with respect to the identified construct using a pre-established rating scale. The ratings of a construct, with respect to each entity in the training set, form a row in the repertory grid. The interview segment is repeated with a different triad of entities, selected randomly from the training set, to elicit the constructs and ratings. The process is repeated until the person is satisfied that there are no more constructs. The interviewee can also volunteer a new construct and rate the entire training set on this construct. He can also add new entities to enlarge the training set. In this case, the new entities are rated on all the previously elicited constructs.

The rows and columns of the repertory grid constitute the complete operational definition of the entities in a person's universe of discourse. Therefore, it can be concluded that two constructs are *functionally identical* if they are assigned the same ratings with respect to all the corresponding entities.

3 Repertory Grid Analysis - A Probabilistic Approach

The repertory grid contains data to approximate the probability distribution of each construct with respect to the entities in the training set. The probability distribution can be used to determine the extent of relevance of each construct in the description of an entity. The raw data in the repertory grid can also be analyzed to quantify and display the extent of mutual relationship between constructs.

There are a number of techniques for repertory grid analysis [21]. The most notable of them include: logic of confirmation analysis [12]; multidimensional scaling [13]; and sum-of-differences [14]. In this section, we describe a technique to improve the analysis stage of a repertory grid by constructing a *dependence tree* using feature dependencies [8]. The resulting dependence tree has a number of advantages over the earlier proposals and is particularly suitable for information retrieval [24].

3.1 Extraction of Relationships Between Constructs

The repertory grid contains the implicit information about the criteria employed by a user in performing some actions. However, this information is obscured by detail and therefore, it is not easy to see the extent of relationship between different constructs.

This excessive information motivates the need for a compact structure, such as a maximum weight dependence tree, that can readily show these relationships. In this subsection, we describe a technique to reduce the repertory grid to a maximum weight dependence tree that displays the mutual relationship between constructs.

Let c_{j_1} and c_{j_2} represent two constructs, corresponding to the rows j_1 and j_2, respectively, in the repertory grid. Let $P(c_j) = u$ be the *a priori* probability of the construct c_j being assigned the rating u and $P(c_{j_1} = u, c_{j_2} = v)$ be the joint probability of the construct c_{j_1} being assigned a rating u when the construct c_{j_2} is assigned a rating v. The extent of mutual dependence between c_{j_1} and c_{j_2} is measured by I, the *expected mutual information measure* (EMIM) [8, 23], and is quantified by

$$I(c_{j_1}, c_{j_2}) = \sum_{u,v} P(c_{j_1} = u, c_{j_2} = v) \log \frac{P(c_{j_1} = u, c_{j_2} = v)}{P(c_{j_1} = u) \cdot P(c_{j_2} = v)} \tag{1}$$

where u and v vary between the two extremes of the rating scale used to elicit the repertory grid. It may be noted that $I(c_{j_1}, c_{j_2}) = I(c_{j_2}, c_{j_1})$ implying that the EMIM is symmetric [23]. Moreover, it can be easily seen that when c_{j_1} and c_{j_2} are independent, $P(c_{j_1}) \cdot P(c_{j_2}) = P(c_{j_1}, c_{j_2})$ for any values of c_{j_1} and c_{j_2} resulting in $I(c_{j_1}, c_{j_2}) = 0$.

The data in a repertory grid can be used to calculate the EMIM between each pair of constructs using Expression 1. Since the EMIM is symmetric, this calculation results in a triangular matrix, called a *similarity matrix* [4]. The similarity matrix displays the quantification of mutual relationship between constructs. In graph-theoretic terms, a similarity matrix corresponds to a complete undirected weighted graph without any loops (reflexive edges). A maximum weight spanning tree of this graph can be used to identify the most important relationships.

3.2 Dependence Tree Representation of Construct Relationships

The similarity matrix captures the dependence relations identified from the repertory grid. In the graph representation of the similarity matrix, each node corresponds to a construct. The weight of an arc is given by the EMIM between the two constructs that define the nodes of the arc.

From the complete graph, a dependence tree can be identified such that the total weight of the arcs in the dependence tree is optimal in some well-defined sense. Such a tree is referred to as the *maximum weight dependence tree.* The maximum weight dependence tree brings out the significant relationship between constructs perceived by the user. It readily displays the dependence of a construct upon a few strongly related constructs; the dependence of remaining constructs can be approximated through a chain of intermediate constructs in the tree.

Let $\mathcal{T}$ be the set of all possible spanning trees that can be derived from the graph representation of the similarity matrix. A maximum weight dependence tree is a spanning tree T such that for all trees $T' \in \mathcal{T}$,

$$\sum_{j=1}^{m} I_T(c_j, c_{N(j)}) \geq \sum_{j=1}^{m} I_{T'}(c_j, c_{N(j)}) \tag{2}$$

where m is the number of nodes (constructs) and $N(\cdot)$ is a function mapping the construct c_j onto its neighbor in the spanning tree. The maximum weight dependence tree can be easily computed from the complete graph by adapting a standard minimum spanning tree algorithm, *e.g.*, Kruskal's algorithm.

The maximum weight dependence tree provides the best approximation to the probability distribution of pairwise mutual dependence between constructs. Let X be a random variable representing the entities. Let $P_T(X)$ be the probability distribution of the random variable on the basis of dependence tree T and $P(X)$ be the actual distribution of X. Chow and Liu [8] showed that the probability distribution $P_T(X)$ is an optimal approximation to $P(X)$ if and only if its dependence tree T has maximum possible weight.

The extent of mutual dependence between two nodes c_{j_1} and c_{j_2} in T is given by the normalized distance between them. The *normalized distance* between the nodes c_{j_1} and c_{j_2} is the total distance between the two nodes along the arcs of T divided by the number of arcs between the two nodes. The conditional probability of degree of dependence between the constructs c_{j_1} and c_{j_2} can be determined by tracing the path between the nodes c_{j_1} and c_{j_2}.

The maximum weight dependence tree captures the dependencies needed to obtain an approximation of the distribution of the random variable X, representing the choice of an entity by the user. The distribution of random variable X is quantified by

$$P_T(X) = \prod_{j=1}^{m} P(c_j | c_{N(j)}) \quad 0 \leq N(j) < j \tag{3}$$

where $\pi = (j_1, j_2, \ldots, j_m)$ is a permutation of the integers $1, 2, \ldots m$, and $N(\cdot)$ is a function mapping j into its neighbor in the maximum weight dependence tree. It is assumed that the end condition $j_1 = 0$. The choice of π depends on the structure of the dependence tree through the function $N(\cdot)$. All permutations of a dependence tree are equivalent in the sense that as long as π is chosen appropriately, the result should not be affected.

4 Cluster Identification and Characterization

The user-oriented clustering schemes proposed by Deogun and Raghavan [10, 19] and Yu, *et al.*, [26, 27] identify membership of a document into a cluster without using document description [9]. These schemes identify the relevance of a cluster of documents to a query by comparing a document from the cluster to the query. The effectiveness of the clustering process can be enhanced if each cluster has a representation that indicates the information content of its documents [3].

During the knowledge elicitation interview, the user identifies a set of concepts that are jointly relevant to the description of a set of documents. Such concepts are dependent on each other, are distinct from the other elicited concepts, and can be collected as a *cluster of concepts*. The documents that are relevant to a concept in a cluster of concepts are generally relevant to the query using any subset of concepts in the cluster. These documents should be assigned to the same cluster of documents. Thus, we have a group of concepts such that if a document is relevant to any concept in the group, it is also relevant to other concepts in the group. Our approach to document clustering is based on the identification of the clusters of concepts. Given a cluster, a user can perceive it as a cluster of concepts while the system views it as a cluster of documents.

The clustering process is based on the development of a *cluster representation* for each cluster. This representation is developed by an expert user by identifying clusters in concepts elicited during the interview. These clusters of concepts are identified by using the maximal weight dependence tree.

The development of clusters of concepts through an interview with the user is known as *laddering* [6]. Laddering involves an interview to refine the elicited knowledge. Laddering identifies the patterns in the knowledge structure (maximum weight dependence tree) developed from the raw knowledge (repertory grid).

The maximum weight dependence tree retains the strongest relationships between concepts. Therefore, it facilitates easy identification of the natural associations between concepts in the user's viewpoint. The user is asked to identify the concepts in the dependence tree that are close to each other to form clusters. After identification of clusters of concepts, the user is asked to specify a concept in each cluster that can be used to represent the entire cluster. This concept is known as a *super-ordinate concept*. Once the concepts have been clustered, and a super-ordinate concept identified, a characterization can be developed for each cluster.

5 Development of Classification Rules

This section describes the development of classification rules using the elicited concepts. The concepts are aggregated into classes and rules are developed to assign the documents to different classes.

Every object has a set of associated attributes that can be used to represent it. These attributes are known as the *primitive attributes* of the object. Within the domain of information retrieval, the primitive attributes are keywords used to describe the documents.

To classify new documents, an appropriate representation must be developed for each concept as well as each cluster. The representation for concepts is developed by determining the *discriminating terms* for the concepts from the document representation. A term is said to be discriminating if its presence in a document representation can indicate the addressability of a concept.

For a given concept, two documents in the training set belong to an equivalence class if and only if they are both jointly relevant (or non-relevant) to that concept. The equivalence classes for each concept are identified from the repertory grid by selecting a threshold τ. Documents whose rating r_{ij} is higher than τ for the concept c_j, belong to the relevant equivalence class δ_j while the other documents belong to the non-relevant equivalence class δ'_j.

$$\begin{array}{rcl} \delta_j & = & \bigcup\{d_i \mid r_{ij} \geq \tau\} \\ \delta'_j & = & \bigcup\{d_i \mid r_{ij} < \tau\} \end{array} \tag{4}$$

The set of equivalence classes for all the concepts is used to develop the equivalence classes of documents corresponding to clusters of concepts. The equivalence class that contains all the documents relevant to a cluster of concepts $\mathcal{C}_k$ is denoted by Δ_k. Δ_k is formed by selecting each document that is relevant to some concept in the corresponding cluster of concepts. Similarly, Δ'_k contains all the documents that are not relevant to any concept in the corresponding cluster of concepts. If D denotes the set of documents $\{d\}$ in the training set then the equivalence classes of documents can be represented by

$$\begin{array}{rcl} \Delta_k & = & \bigcup\{d_i \mid d_i \in \delta_j; \; c_j \in \mathcal{C}_k\} \\ \Delta'_k & = & D - \Delta_k \end{array} \tag{5}$$

The characterization for each cluster of concepts $\mathcal{C}_k$ is developed from the representation of documents in the training set. The indexing is performed using the options of term frequency (TF) and inverse document frequency (IDF) weights, stemming of keywords, and ignoring the common words through a stop list. The selected keywords are then processed through a filter to remove the keywords with a weight less than a prespecified threshold θ. Document d_i is thus represented as a set of index terms as follows:

$$d_i = \{t_{i_1}, t_{i_2}, \ldots\} \tag{6}$$

The representation for an equivalence class is developed by using the discriminating terms for the documents in that class. These terms are present only in the documents that are in the equivalence class. If the representation for a group of concepts $\mathcal{C}_k$ is denoted by the pair (G_k, G'_k), we have

$$\begin{array}{l} G_k = \cup\{t_i\} - \cup\{t'_i\} \quad | \; t_i \in d_i,\; d_i \in \Delta_k,\; t'_i \in d'_i,\; d'_i \in \Delta'_k \\ G'_k = \cup\{t'_i\} - \cup\{t_i\} \quad | \; t_i \in d_i,\; d_i \in \Delta_k,\; t'_i \in d'_i,\; d'_i \in \Delta'_k \end{array} \tag{7}$$

This gives us a system level representation for groups of concepts such that the concepts in a group are judged to be similar. Therefore, the documents addressing these concepts should be assigned to the same cluster.

The classification rules are developed as follows. Let $\circ$ denote a *matching operator* and d be a document to be classified represented by index terms. The document is matched against the representation of each concept group and is assigned to the group for which the value of match is maximized.

$$d \in \mathcal{C}_{k'} \;\Leftarrow\; (d \circ G_{k'_j} - d \circ G'_{k'_j}) = \max_{\forall \mathcal{C}_k} \; (d \circ G_{k_j} - d \circ G'_{k_j}) \tag{8}$$

The matching operator $\circ$ in Expression 8 can be any standard coefficient from Table 1 [23]. Using the operator, the relevance of a document to each cluster is evaluated. The document is assigned to the cluster for which this match is maximized.

Simple matching coefficient	$\lvert X \cap Y \rvert$
Dice's coefficient	$2 \cdot \frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$
Jaccard's coefficient	$\frac{\lvert X \cap Y \rvert}{\lvert X \cup Y \rvert}$
Cosine coefficient	$\frac{\lvert X \cap Y \rvert}{\sqrt{\lvert X \rvert \times \lvert Y \rvert}}$
Overlap coefficient	$\frac{\lvert X \cap Y \rvert}{\min(\lvert X \rvert, \lvert Y \rvert)}$

Table 1: Different coefficients of match

6 Experimental Results

Several experiments were conducted to elicit the opinion of users of the relevance of a sample of documents through a knowledge acquisition program. The experiments were conducted using four subjects and three bibliographies. The experiments were limited to relatively small collections due to the fact that it was difficult to find volunteers who could read the documents in a bibliographic collection and assign a rating to each of them for evaluation.

The collection employed in the experiment described here consists of 205 documents in the `refer` format [22]. Each entry in the bibliography consists of a complete citation and an abstract. The abstracts in the bibliography are taken from the document itself. In some cases, where an abstract was not available, the abstract was selected from a standard review source, for example *Computer and Control Abstracts*.

The bibliography was developed as a part of the evaluation of this work. Since the system is based on a highly personal environment, it is essential for system evaluation that each document in the bibliography be known to the user. This requirement is imposed only for the evaluation of results and is not a constraint in an operational information retrieval system based on the techniques developed.

Personal construct theory requires that the interviewee be familiar with all the elements in the training set. To achieve this objective, the user is presented with a few documents in the collection and asked to select a training set such that the documents in the training set adequately represent his information requirements[1]. This is necessary because the system must have at least one example of a relevant docu-

[1] In some cases, the user may not explicitly remember to include documents that may be relevant in the description of his viewpoint. In such a case, the user is allowed to add more documents at the end of the interview.

ment for each concept employed by the user to indicate his information requirements. However, it was experimentally observed that to develop a good representation, the training set must contain between three to five examples for each concept.

The training set was made of 24 documents. To make this selection, the system presents the documents one-by-one and asks the user whether the document is well-known to him to be included in the training set. However, in a large system, this approach has obvious flaws. Therefore, it is proposed that the user be allowed to select his own set of documents that satisfies the conditions for the training set.

During the knowledge elicitation interview, the system selects three documents from this training set and presents them to the user. The user distinguishes between these documents such that at least two of the documents address a common topic. If two documents do not address a common topic, the user can specify a topic that is addressed in one document but not in the other two. This topic forms one concept (construct). The user is asked to assign a rating to all the documents in the training set on this concept. The rating scale in our experiments ranged from 1 to 5 with a 1 indicating that the document is not at all relevant to the concept and a 5 indicating that the document is relevant to the concept.

The interview is repeated with different triads from the training set to elicit other concepts and ratings are assigned to all documents with respect to these concepts. At any stage during the interview, the user is free to volunteer some concepts and ratings of all documents on those concepts are elicited as well. The complete set of ratings elicited from the user is collected in a repertory grid. An example of such a grid is presented in Tables 2-3. This grid was elicited using a training set of 24 documents from the overall collection. During the interview, 21 concepts (Table 4) were elicited from the user to describe his interests in the documents in the training set.

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}
c_1	4	5	4	5	2	2	5	5	5	5	5	5
c_2	3	2	3	4	4	5	2	3	5	5	2	5
c_3	1	1	1	1	1	1	1	1	1	1	1	1
c_4	2	2	2	1	1	1	3	5	1	1	5	1
c_5	1	1	1	1	1	1	1	1	1	1	1	1
c_6	1	1	1	3	1	1	1	1	3	1	1	1
c_7	2	1	2	4	5	4	1	2	3	2	1	2
c_8	1	1	1	4	1	2	2	2	4	2	1	1
c_9	1	1	1	1	1	1	1	1	1	1	1	1
c_{10}	1	4	1	4	1	1	4	2	3	2	1	1
c_{11}	5	3	5	3	2	2	3	2	3	3	2	2
c_{12}	1	1	1	1	1	1	1	1	1	1	1	1
c_{13}	1	1	1	1	1	1	3	1	1	1	1	1
c_{14}	2	3	2	2	1	1	5	2	3	3	2	1
c_{15}	2	3	2	4	5	5	2	1	4	4	1	5
c_{16}	1	1	1	1	1	1	1	1	1	1	1	1
c_{17}	1	5	1	5	2	2	4	2	5	4	2	1
c_{18}	2	5	2	5	1	1	4	3	5	4	1	1
c_{19}	1	1	1	1	1	1	1	5	1	1	5	1
c_{20}	1	3	1	2	1	1	3	5	2	1	5	1
c_{21}	1	2	1	1	1	1	3	2	2	2	2	1

Table 2: Repertory grid of the expert's evaluation of documents 1-12

	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}	d_{20}	d_{21}	d_{22}	d_{23}	d_{24}
c_1	3	5	1	1	1	1	1	1	3	1	1	1
c_2	1	5	1	3	3	2	2	3	5	1	1	1
c_3	1	1	1	5	5	5	5	5	2	1	1	1
c_4	5	1	5	5	5	2	5	3	2	5	4	5
c_5	1	1	5	1	1	1	1	1	1	5	5	5
c_6	1	1	1	1	1	1	1	1	5	1	1	1
c_7	1	2	1	1	1	2	1	1	3	1	1	1
c_8	1	3	2	5	5	4	5	5	2	1	2	1
c_9	1	1	5	1	1	1	1	1	1	3	5	5
c_{10}	1	4	3	5	5	4	5	4	3	2	3	2
c_{11}	1	3	1	1	1	1	1	1	2	1	1	1
c_{12}	1	1	5	1	1	1	1	2	1	5	5	5
c_{13}	1	1	5	1	1	1	1	1	1	4	5	5
c_{14}	1	3	5	4	4	3	4	4	1	1	5	2
c_{15}	1	4	1	1	1	2	1	1	4	1	1	1
c_{16}	1	1	1	5	5	5	5	5	2	1	1	1
c_{17}	1	5	4	5	5	4	5	5	2	1	5	3
c_{18}	1	4	5	5	5	4	5	5	2	3	5	4
c_{19}	4	1	5	5	5	4	5	2	1	5	4	4
c_{20}	2	2	1	4	4	1	4	2	2	2	1	1
c_{21}	1	2	4	5	4	3	4	5	2	3	4	4

Table 3: Repertory grid of the expert's evaluation of documents 13-24

The repertory grid is analyzed using the EMIM (Expression 1) to determine the pairwise dependence of each possible pair of constructs. The resulting similarity matrix is presented in Tables 5-6.

The similarity matrix corresponds to a complete undirected weighted graph, with the EMIM $I(c_{j_1}, c_{j_2})$ as the weight of the arc between the constructs c_{j_1} and c_{j_2}.

c_1	information retrieval	c_2	classification
c_3	personal construct theory	c_4	expert systems
c_5	intelligent tutoring systems	c_6	rough sets
c_7	graph theory	c_8	knowledge acquisition
c_9	programming tutors	c_{10}	machine learning
c_{11}	hypertext	c_{12}	education training
c_{13}	student models	c_{14}	user profile
c_{15}	cluster analysis	c_{16}	repertory grid
c_{17}	adaptive systems	c_{18}	user-oriented systems
c_{19}	rule-based systems	c_{20}	fuzzy logic
c_{21}	psychology		

Table 4: Constructs elicited from the user

	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}
c_1	-	0.62	0.47	0.63	0.28	0.26	0.64	0.42	0.28	0.57	0.97
c_2		-	0.33	0.71	0.41	0.25	0.68	0.46	0.41	0.49	0.59
c_3			-	0.26	0.10	0.20	0.25	0.49	0.10	0.44	0.36
c_4				-	0.27	0.24	0.63	0.43	0.28	0.53	0.59
c_5					-	0.05	0.21	0.13	0.45	0.31	0.25
c_6						-	0.37	0.30	0.05	0.23	0.24
c_7							-	0.41	0.21	0.44	0.52
c_8								-	0.17	0.79	0.48
c_9									-	0.34	0.25
c_{10}										-	0.55
c_{11}											-
c_{12}											
c_{13}											
c_{14}											
c_{15}											
c_{16}											
c_{17}											
c_{18}											
c_{19}											
c_{20}											
c_{21}											

Table 5: Expected Mutual Information Measures (Constructs 1-11)

	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}	c_{20}	c_{21}
c_1	0.35	0.30	0.72	0.75	0.47	0.54	0.64	0.55	0.56	0.79
c_2	0.46	0.47	0.70	0.80	0.33	0.37	0.39	0.58	0.49	0.65
c_3	0.13	0.12	0.46	0.23	0.68	0.28	0.31	0.29	0.37	0.37
c_4	0.38	0.40	0.57	0.92	0.26	0.26	0.52	0.78	0.52	0.57
c_5	0.45	0.45	0.19	0.23	0.10	0.16	0.14	0.24	0.14	0.31
c_6	0.06	0.06	0.16	0.30	0.20	0.20	0.24	0.16	0.25	0.15
c_7	0.26	0.26	0.54	0.74	0.25	0.40	0.50	0.43	0.45	0.59
c_8	0.18	0.24	0.83	0.47	0.49	0.74	0.54	0.34	0.53	0.54
c_9	0.45	0.55	0.28	0.23	0.10	0.24	0.23	0.27	0.22	0.41
c_{10}	0.35	0.39	0.74	0.60	0.44	0.69	0.93	0.41	0.70	0.75
c_{11}	0.32	0.29	0.70	0.83	0.36	0.69	0.68	0.59	0.51	0.68
c_{12}	-	0.45	0.26	0.29	0.13	0.19	0.20	0.39	0.19	0.43
c_{13}		-	0.41	0.29	0.12	0.35	0.32	0.28	0.33	0.53
c_{14}			-	0.64	0.46	0.77	0.72	0.37	0.61	0.78
c_{15}				-	0.23	0.46	0.66	0.65	0.66	0.76
c_{16}					-	0.28	0.31	0.29	0.37	0.37
c_{17}						-	0.89	0.23	0.56	0.65
c_{18}							-	0.43	0.46	0.66
c_{19}								-	0.51	0.53
c_{20}									-	0.57
c_{21}										-

Table 6: Expected Mutual Information Measures (Constructs 12-21)

Therefore, it can be used to identify a maximum weight dependence tree by applying Kruskal's algorithm. The arcs chosen by Kruskal's algorithm are presented in Table 7. The dependence tree is presented in Figure 1.

6.1 Cluster Characterization and Assignment

The development of the maximum weight dependence tree is followed by laddering. The maximum weight dependence tree is presented to the user who highlights the clusters of concepts in the tree. The resulting dependence tree with clusters enclosed in polygons is presented in Figure 2. The four superordinate concepts are "knowledge acquisition," "intelligent tutoring systems," "information retrieval," and "expert systems" and are typed in boldface capitals.

Representation for each cluster of concepts was developed by using the repertory grid (Tables 2-3). The documents ranked by the user as $\geq \tau$ ($\tau = 3$) were deemed to be relevant to the concepts. This partitioning of documents is used to create an equivalence class of documents (δ_j, δ'_j) with respect to each concept. The equiva-

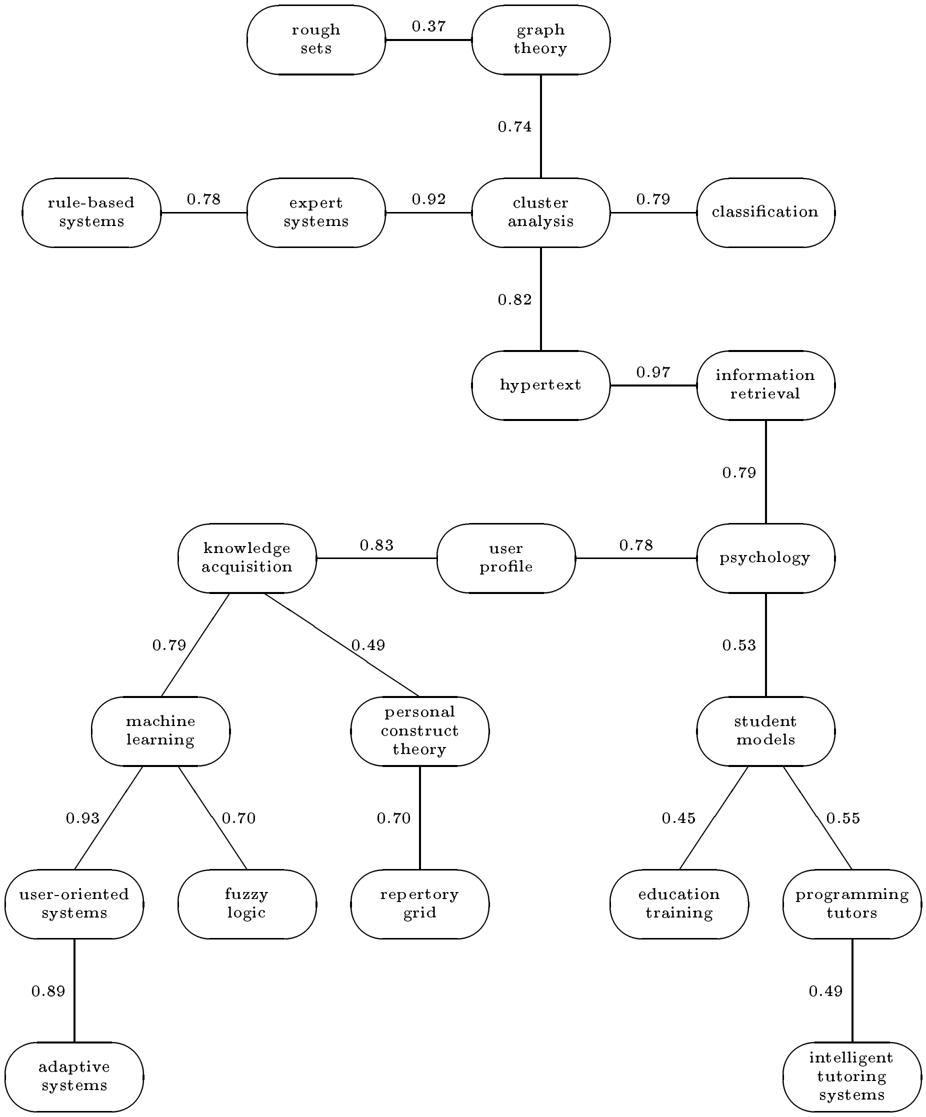

Figure 1: Maximum weight dependence tree

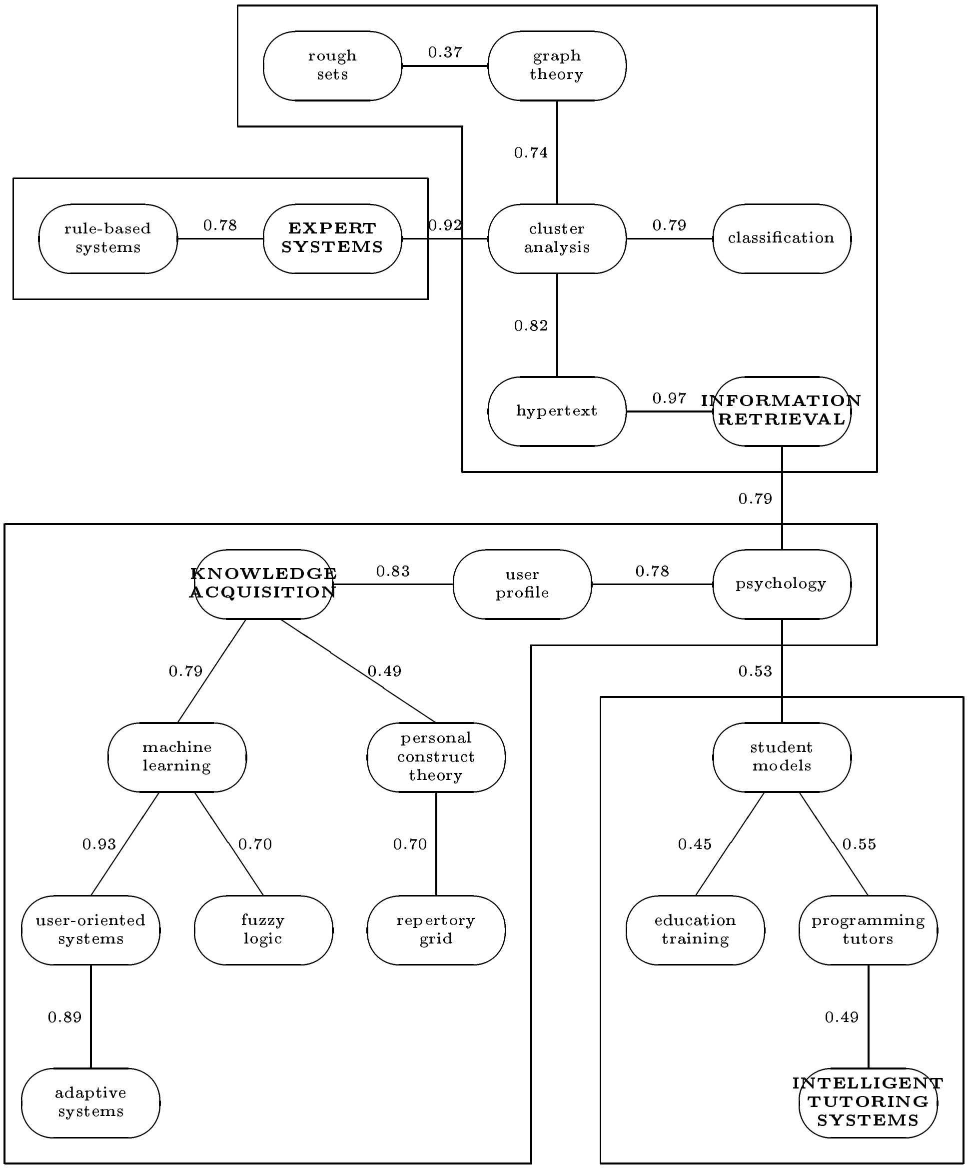

Figure 2: Maximum weight dependence tree for constructs after laddering

Concept c_{j_1}		Concept c_{j_2}		EMIM
c_1	information retrieval	c_{11}	hypertext	0.97
c_{10}	machine learning	c_{18}	user-oriented systems	0.93
c_4	expert systems	c_{15}	cluster analysis	0.92
c_{17}	adaptive systems	c_{18}	user-oriented systems	0.89
c_8	knowledge acquisition	c_{14}	user profile	0.83
c_{11}	hypertext	c_{15}	cluster analysis	0.82
c_1	information retrieval	c_{21}	psychology	0.79
c_2	classification	c_{15}	cluster analysis	0.79
c_8	knowledge acquisition	c_{10}	machine learning	0.79
c_4	expert systems	c_{19}	rule-based systems	0.78
c_{14}	user profile	c_{21}	psychology	0.78
c_7	graph theory	c_{15}	cluster analysis	0.74
c_{10}	machine learning	c_{20}	fuzzy logic	0.70
c_3	personal construct theory	c_{16}	repertory grid	0.68
c_9	programming tutors	c_{13}	student models	0.55
c_{13}	student models	c_{21}	psychology	0.53
c_3	personal construct theory	c_8	knowledge acquisition	0.49
c_5	intelligent tutoring systems	c_9	programming tutors	0.49
c_{12}	education training	c_{13}	student models	0.45
c_6	rough sets	c_7	graph theory	0.37

Table 7: Constructs with maximal information measure

lence classes and the keyword representation for each document, determined through SMART system, were used to develop the representation (C_j, C'_j) for each concept.

The sets of documents δ_j and δ'_j (Expression 4) and their representation are identified from the repertory grid to develop cluster characterization. Using the representation, the documents with respect to each cluster of concepts, (Δ_k, Δ'_k), are identified (Expression 5). While developing the actual representation for each cluster $\mathcal{C}_k$, the index terms with a TF$\times$IDF weight of less than 2.0 are ignored to eliminate the keywords with low weight [11]. The actual representation (G_k, G'_k) for each cluster $\mathcal{C}_k$ is developed by using Expression 7.

In the first experiment, a collection of 122 documents was classified. The training set was selected from this collection. The documents were classified by using the rules in Expression 8 and using different matching functions (Table 1). The approach used for validation is to check the efficacy of the procedure in actual classification.

To evaluate the classification, the user was presented with the entire collection prior to the experiment and asked to express his opinion about the most appropriate class, from the set of classes already elicited, to which each document belongs. This opinion is later used to evaluate whether the documents are correctly classified.

Let POS be the number of possible documents in a cluster, ACT be the actual number of documents assigned to a cluster, COR be the number of documents correctly assigned, and SPU be the number of documents spuriously assigned. Then, the clustering can be evaluated using the measures of precision and recall given by

$$\text{Precision} = \frac{\text{COR}}{\text{ACT}} \tag{9}$$

$$\text{Recall} = \frac{\text{COR}}{\text{POS}} \tag{10}$$

Another measure of evaluation is given by *overgeneration* [16]. Overgeneration is the ratio of the number of spurious documents in a cluster to actual number of documents in the cluster:

$$\text{Overgeneration} = \frac{\text{SPU}}{\text{ACT}} \tag{11}$$

The representation of each document is matched against the representation of clusters of constructs and this matching is quantified. The document is assigned to the cluster for which this matching is maximized. Our initial experiments were conducted with a collection of 122 documents and it was determined that the cosine matching coefficient provided the best results. This is because the cosine coefficient takes into account the cardinality of the cluster representations as well as the cardinality of the document representation. Hence, the discussion of the experiments largely focuses on the cosine match. Table 8 shows that more than 80% of the documents are correctly classified. 4% of the documents are not classified because of inadequate number of classes. This is expected as these documents are not typical of any class in the collection and were not represented by any document in the training set. About 14% of the documents are classified incorrectly. However, the experiments showed that the documents exhibit only a small difference in the value of matching coefficients between correct and assigned classes.

	Possible	Actual	Correct	Spurious	Recall	Precision	Overgeneration
IR	44	54	40	10	0.91	0.74	0.19
KA	24	31	23	7	0.96	0.74	0.23
ES	26	26	26	0	1.00	1.00	0.00
ITS	10	11	10	1	1.00	0.91	0.09
Overall	104	122	99	18	0.95	0.85	0.15

Table 8: Clustering of a collection of 122 documents using cosine matching coefficient

The evaluation of results of clustering for the second collection (158 documents) and the third collection (205 documents) are presented in Tables 9 and 10, respectively. In Table 10, the number of spurious documents is high because the new documents added to the collection did not belong to any of the clusters, affecting the precision in the process.

	Possible	Actual	Correct	Spurious	Recall	Precision	Overgeneration
IR	45	65	44	21	0.98	0.68	0.32
KA	66	64	53	11	0.80	0.83	0.17
ES	18	17	15	2	0.83	0.88	0.11
ITS	15	11	10	1	0.67	0.91	0.09
Overall	142	157	122	35	0.86	0.78	0.22

Table 9: Evaluation of clustering in the collection of 158 documents

	Possible	Actual	Correct	Spurious	Recall	Precision	Overgeneration
IR	57	86	47	39	0.82	0.55	0.45
KA	101	85	66	19	0.65	0.78	0.22
ES	18	23	15	8	0.83	0.65	0.35
ITS	15	11	10	1	0.67	0.91	0.09
Overall	191	205	138	67	0.70	0.67	0.33

Table 10: Evaluation of clustering in the collection of 205 documents

7 Discussion

The user-oriented clustering methods [26, 27] and the learning automaton [18] are examples of general-to-specific search with a strategy for queries [7]. In contrast, our approach uses specific-to-general search under the *justifiability assumption* [7]. The justifiability assumption states that sufficient knowledge is supplied by the user from which valid generalizations of seed facts can be justified and constructed.

The clustering technique described in this paper satisfies all the six conditions for good clusters [26]. As required by the first condition, closely related terms are assigned to the same clusters. This is evident as the assignment is performed by the user through the laddering process. Moreover, documents addressing the same topic are assigned to the same cluster (Table 9). Some documents may be assigned to wrong classes if there is no document in the training set corresponding to their topic. The second condition dictates that the system should not collect statistics on queries. Our clustering is independent of querying process as its basis lies in the knowledge elicited from an expert user. Since the query process is independent, no documents or files are moved as a result of queries thus satisfying the third condition. To satisfy the fourth condition, it is observed that the clustering is based on the development of a cluster representation. It is only in the training set that individual concepts are compared to each other. Once the cluster representation is developed, new documents are compared to the representation of each cluster to decide its assignment. Thus a document is assigned to an appropriate cluster in polynomial time. The fifth requirement suggested that the clustering scheme should allow dependencies so as to retrieve a number of dependent documents. To conform with this requirement, queries can be matched against cluster representations to select or ignore a group of documents as a whole. Finally, the clustering scheme is conceptually simple. The scheme calls for the development of cluster representations based on a knowledge acquisition interview and the documents are assigned to clusters using this representation.

The technique is suitable for interview by a single user who may express his opinion taking into account the requirements of several users, for example a librarian. This is because of the nature of the personal construct theory which forms the cornerstone of this technique.

The main advantage of our technique is that it permits a quick transfer to automatic classification from an existing manual system. The existing manual classification can be preserved with minimal disruption and minimal setup time. The examples for the training set can be selected from the existing clusters thus minimizing the development effort. Even if a manual system is not in place, the rules for classification are developed by observing some examples provided by the user rather than through a complex rule-formulation process.

References

[1] M. R. Anderberg. *Cluster Analysis for Applications.* Academic Press, New York, NY, 1973.

[2] S. K. Bhatia, J. S. Deogun, and V. V. Raghavan. Formation of Categories in Document Classification Systems. In N. A. Sherwani, E. de Doncker, and J. A. Kapenga, editors, *Computing in the 90's: The First Great Lakes Computer Science Conference*, pages 91–97, Kalamazoo, MI, October 1989. Springer-Verlag. Lecture Notes in Computer Science # 507.

[3] S. K. Bhatia, J. S. Deogun, and V. V. Raghavan. Assignment of Term Descriptors to Clusters. In *Proceedings of the 1990 Symposium on Applied Computing*, pages 181–185, Fayetteville, AR, April 1990. IEEE Computer Society Press.

[4] S. K. Bhatia, J. S. Deogun, and V. V. Raghavan. Conceptual query formulation and retrieval. *Journal of Intelligent Information Systems*, 5(3):183–209, November 1995.

[5] J. H. Boose. A Knowledge Acquisition Program for Expert System Based on Personal Construct Theory. *International Journal of Man-Machine Studies*, 23:495–525, 1985.

[6] J. H. Boose. *Expertise Transfer for Expert System Design.* Elsevier-Science Publishers, New York, 1986.

[7] W. Buntine. Induction of Horn Clauses: Methods and the Plausible Generalization Algorithm. In B. Gaines and J. Boose, editors, *Knowledge Acquisition for Knowledge-Based Systems (vol. 1)*, pages 277–297. Academic Press, San Diego, CA, 1988.

[8] C. K. Chow and C. N. Liu. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Transactions on Information Theory*, IT-14:462–467, 1968.

[9] J. S. Deogun, S. K. Bhatia, and V. V. Raghavan. Automatic Cluster Assignments for Documents. In *Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications*, Miami Beach, FL, February 1991.

[10] J. S. Deogun and V. V. Raghavan. User-Oriented Document Clustering: A Framework for Learning in Information Retrieval. In *Proceedings of the ACM SIGIR Conference*, pages 157–163, Pisa, Italy, September 1986.

[11] J. S. Deogun and V. V. Raghavan. Description of the UNL/USL System Used for MUC-3. In B. Sundheim, editor, *Proceedings of the Third Message Understanding Conference*, San Diego, CA, June 1991.

[12] K. Ford and F. Petry. Knowledge Acquisition from Repertory Grids Using A Logic of Confirmation. *SIGART Newsletter*, (108):146–147, April 1989.

[13] J. G. Gammack. Different Techniques and Different Aspects on Declarative Knowledge. In A. L. Kidd, editor, *Knowledge Acquisition for Expert Systems: A Practical Handbook*, pages 137–163. Plenum Press, New York, NY, 1987.

[14] A. Hart. *Knowledge Acquisition for Expert Systems.* McGraw-Hill, New York, NY, 1986.

[15] G. A. Kelly. *The Psychology of Personal Constructs.* Norton Publishers, New York, NY, 1955.

[16] Naval Oceanographic Systems Center. MUC Scoring System: User's Manual, April 1991. Scoring System for the Third Message Understanding Conference, San Diego, CA : Version 2.1 : Sponsored by DARPA/NOSC.

[17] B. Nordhausen. Conceptual Clustering Using Relational Information. In *AAAI-86: Proceedings of the Fifth National Conference on Artificial Intelligence*, pages 508–512, Philadelphia, PA, August 1986. (volume 1).

[18] B. J. Oommen and D. C. Y. Ma. Fast Object Partitioning Using Stochastic Learning Automata. In *Proceedings of the Tenth ACM SIGIR*, pages 111–122, New Orleans, LA, 1987.

[19] V. V. Raghavan and J. S. Deogun. Optimal Determination of User-Oriented Clusters. In *Proceedings of the Tenth ACM SIGIR*, pages 140–146, New Orleans, LA, 1987.

[20] M. L. G. Shaw. Knowledge Support Systems. In P. A. Luker and G. Birtwistle, editors, *Simulation and AI*, pages 58–63. Society for Computer Simulation, San Diego, CA, July 1987. Simulation Series, vol. 18, no. 3.

[21] M. L. G. Shaw and B. R. Gaines. An Interactive Knowledge-Elicitation Technique Using Personal Construct Technology. In A. L. Kidd, editor, *Knowledge Acquisition for Expert Systems: A Practical Handbook*, pages 109–136. Plenum Press, New York, NY, 1987.

[22] B. Tuthill. Refer – A Bibliography System. In *Unix Programmer's Manual*. Bell Laboratories, Murray Hill, NJ, 1978.

[23] C. J. van Rijsbergen. *Information Retrieval*. Butterworth Publishers, Boston, MA, 2nd edition, 1981.

[24] C. J. van Rijsbergen, D. J. Harper, and M. F. Porter. The Selection of Good Search Terms. *Information Processing and Management*, 17:77–91, 1981.

[25] E. M. Voorhees. The Cluster Hypothesis Revisited. In *Proceedings of the Eighth ACM SIGIR*, pages 188–196, Montreal, Quebec, Canada, 1985.

[26] C. T. Yu, M. K. Siu, K. Lam, and F. Tai. Adaptive Clustering Schemes: General Framework. In *IEEE COMPSAC 81: Proceedings of the IEEE Computer Society's Fifth International Computer Software and Applications Conference*, pages 81–89, Chicago, IL, November 1981.

[27] C. T. Yu, Y. T. Wang, and C. H. Chen. Adaptive Document Clustering. In *Proceedings of the Eighth ACM SIGIR*, pages 197–203, Montreal, Quebec, Canada, 1985.