In: Wu, W., Xiong, H. and Shekhar, S. (Eds.),
Information Retrieval and Clustering
Kluwer Academic Publishers, pp. 299-329, 2003.

Granular Computing for the Design of Information Retrieval Support Systems

Y.Y. Yao
Department of Computer Science
University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: yyao@cs.uregina.ca

Contents

1 Introduction

Information Retrieval (IR) systems were traditionally used in libraries as a tool for searching for relevant information with respect to the user information needs [39]. An IR system is designed with the objective to provide useful and only useful documents from a large document collection [3, 39, 45]. The introduction of the World Wide Web (the Web), digital libraries, as well as many markup languages, has offered new opportunities and challenges to information retrieval researchers [3].

The Web is a totally new media for communication, which goes far beyond other communication medias, such as paper form publishing, radio, telephone and television. It revolutionizes the way in which information is gathered, stored, processed, presented, shared, and used. In this study, we concentrate on one particular use of the Web as a media and tool supporting scientific research. The Web has significant impacts on academic research. In contrast to traditional libraries and paper form publishing, it is a new platform for carrying out scientific research [28]. The amount of scientific information, such as online journals, books and scientific databases, increases in a very fast speed [21]. The existence of many effective tools, such as search engines and online reference services, makes scientific literature immediately accessible to a large group of scientists [21]. Studies have shown that articles available on the Web are more highly cited and used [20, 28]. The Web becomes a large and searchable virtual library. The problem of making effective use of the Web for research is a challenge for every scientist.

A basic tool to support research through the Web is search engines. The

Web search engines are the most heavily-used online services [8]. The effective use of the Web depends on, to a large extent, the success of many search engines [21]. Many Web search engines are designed based on the principle of information retrieval [22]. They inherit many of the disadvantages of traditional IR systems. IR systems focus mainly on the retrieval functionality, namely, the selection of a subset of documents from a large collection. There is little support for other activities of scientific research. IR systems use simple document and query representation schemes. A document is typically represented as a list of keywords, and a query is represented as either a list of keywords or a Boolean expression. There is little consideration of the relationships between different documents and between different portions of the same document. Semantic and structure information in each document is not used. IR systems use simple pattern based matching method to identify relevant documents. The philosophy and technologies of IR may be sufficient to support scientific research in the conventional library environment, where structure and semantic information of documents is not readily available. IR systems are inadequate to support research on the new Web platform.

Many researchers have attempted to extend and modify IR systems to meet the new challenges brought by the Web [3]. For example, research has been done on the use of hyper-text documents [3], structured documents [2], semantic information represented by ontology [15], and automatic citation analysis [23]. The research by Lawrence's group covers many important topics and directions on the use of the Web for supporting scientific research (http://www.neci.nec.com/∼lawrence/papers.html). However, many such existing studies focus on particular supporting functionalities in isolation. One needs to address many interrelated and interactive functionalities in a more general framework.

In this chapter, we introduce the notion of Information Retrieval Support Systems (IRSS) as a general framework for supporting scientific research [56, 58]. IRSS is viewed as the next generation in the evolution of retrieval systems. IRSS is based on a new design philosophy which emphasizes many supporting functionalities, in addition to the simple retrieval functionality. As a more concrete example to demonstrate the potential value of IRSS, we describe a Granular Computing (GrC) model for the organization and retrieval of scientific XML documents [58].

The rest of the chapter is organized as follows. Section 2 reviews basic concepts of granular computing. Section 3 reviews basic concepts of information, with emphasis on granulation of document space, query (user)

space, term space, and retrieval results. In Section 4, we argue that IRSS is the next generation in the evolution of retrieval systems. Section 5 discusses the basic issues of IRSS. Section 6 describes in detail the organization and retrieval of scientific XML documents.

The main objective of this chapter is to draw attention of information retrieval researchers to IRSS. It is hoped that many related, but isolated studies, topics, techniques, tools, and systems can be unified under the umbrella of IRSS.

This chapter summarizes and extends our preliminary studies on IRSS, and draws many results from two recent papers [56, 58].

2 Granular Computing

As a recently renewed research topic, granular computing (GrC) is an umbrella term to cover any theories, methodologies, techniques, and tools that make use of granules (i.e., subsets of a universe) in problem solving [25, 54, 55, 61]. Basic ingredients of granular computing are subsets, classes, and clusters of a universe. They have been considered either explicitly or implicitly in many fields, such as data and cluster analysis, database and information retrieval, concept formation, machine learning, and data mining [53, 61].

2.1 Basic Issues

There are many fundamental issues in granular computing, such as granulation of the universe, description of granules, relationships between granules, and computing with granules. Issues of granular computing may be studied from two related aspects, the construction of granules and computing with granules. The former deals with the formation, representation, and interpretation of granules, while the latter deals with the utilization of granules in problem solving.

Granulation of a universe involves the decomposition of the universe into parts, or the grouping of individual elements into classes, based on available information and knowledge. Elements in a granule are drawn together by indistinguishability, similarity, proximity or functionality [61]. The interpretation of granules focuses on the semantic side of granule construction. It addresses the question of why two objects are put into the same granule. It is necessary to study criteria for deciding if two elements should be put into the same granule, based on available information. One must provide necessary

semantic interpretations for notions such as indistinguishability, similarity, and proximity. It is also necessary to study granulation structures derivable from various granulations of the universe [59]. The formation and representation of granules deal with algorithmic issues of granule construction. They address the problem of how to put two objects into the same granule. Algorithms need to be developed for constructing granules efficiently.

Computing with granules can be similarly studied from both the semantic and algorithmic perspectives. On the one hand, one needs to interpret various relationships between granules, such as closeness, dependency, and association, and to define and interpret operations on granules. On the other hand, one needs to design techniques and tools for computing with granules, such as approximation, reasoning, and inference.

According to Zadeh [61], granular computing suggests the basic guiding principle of fuzzy logic:

> "*Exploit the tolerance for imprecision, uncertainty and partial truth to achieve tractability, robustness, low solution cost and better rapport with reality.*"

It offers a more practical philosophy for real world problem solving. Instead of searching for the optimal solution, one may search for good approximate solutions. One only needs to examine the problem at a finer granulation level with more detailed information when there is a need or benefit for doing so.

It should be pointed out that studies of granular computing are only complementary to vigorous investigations on precise and non-granular computational approaches. The latter may provide justifications and guidelines for the former.

2.2 Simple Granulations

Let U be a finite and non-empty set called the universe, and let $E \subseteq U \times U$ denote an equivalence relation on U. The pair $apr = (U, E)$ is called an approximation space [29]. The equivalence relation E partitions the set U into disjoint subsets. This partition of the universe is called the quotient set induced by E and is denoted by $U/E = \{[x]_E \mid x \in U\}$, where

$$[x]_E = \{y \mid y \in U, xEy\}, \tag{1}$$

is the equivalence class containing x. The equivalence relation is the available information or knowledge about the objects under consideration. It

represents a very special type of similarity between elements of the universe. If two elements $x, y \in U$ belong to the same equivalence class, we say that x and y are indistinguishable, i.e., they are similar. Each equivalence class may be viewed as a granule consisting of indistinguishable elements. It is also referred to as an equivalence granule. The granulation structure induced by an equivalence relation is a partition of the universe. There is a one-to-one correspondence between equivalence relations and partitions of the universe.

A partition is only a very restricted granulation of the universe, in which no overlap between granules is allowed. In general, one can use a covering of the universe to granulate a universe. In this case, a universe is divided into a family of possibly overlap granules. By allowing the overlap between granules, one can put an element into more than one granule.

There is no longer a one-to-one correspondence between coverings of a universe and certain type of binary relations on the universe. Nevertheless, one may still construct a covering using some specific type of binary relations. Suppose that a binary relation $R \subseteq U \times U$ is used to represent the similarity between elements of the universe. It is reasonable to assume that similarity is at least reflexive, but not necessarily symmetric and transitive [41]. For a reflexive binary relation R on U, if xRy, we say that y is R-related to x. A binary relation may be more conveniently represented using successor neighborhoods, or successor granules [53]:

$$(x)_R = \{y \mid y \in U, xRy\}. \tag{2}$$

The successor neighborhood $(x)_R$ consists of all R-related elements of x. When R is an equivalence relation, $(x)_R$ is the equivalence class containing x. When R is a reflexive relation, the family of successor neighborhoods $U/R = \{(x)_R \mid x \in U\}$ is a covering of the universe, namely, $\bigcup_{x \in U}(x)_R = U$.

A partition or a covering can be viewed as a brief description, or coarsened and granulated view, of the universe. We consider each equivalence class in a partition or a class in a covering as a whole instead of many individuals. In general, the number of the equivalence classes or subsets in a covering are much smaller than the number of the elements of the universe. The coarsening of the universe enables us to ignore or omit some fine differences between distinct objects. In other words, the coarsened view may allow us to observe some basic structures of the universe, which may not be easily observable under the very fine description of objects. The coarsening of the universe may also reduce computational costs of operations on the universe.

2.3 Hierarchical Granulations

The simple one-level granulated views of a universe are based on binary relations representing the simplest type of similarities between elements of the universe. Two elements are either related or unrelated. To avoid such a limitation, multi-level granulation structures can be constructed by putting together simple granulation structures [55]. Each level of the complex structure is a simple granulation structure such as a partition or a covering.

A multi-level hierarchical granulation structure can be interpreted and constructed by a nested sequence of binary relations [26, 30, 55]. Recall that a binary relation on U is a subset of the Cartesian product $U \times U$. The set inclusion defines an order on binary relations on U. An equivalence relation E_1 is said to be finer than another equivalence relation E_2, or E_2 is coarser than E_1, if $E_1 \subseteq E_2$. For the corresponding partitions, we write $U/E_1 \preceq U/E_2$. A finer relation produces smaller granules than a coarser relation, i.e., $[x]_{E_1} \subseteq [x]_{E_2}$ for all $x \in U$. Each equivalence granule of E_2 is in fact a union of some equivalence granules of E_1. Each granule of E_1 is obtained by further partitioning a granule of E_2. In general, we may consider a nested sequence of m equivalence relations:

$$E_1 \subseteq E_2 \subseteq \ldots \subseteq E_m. \tag{3}$$

The corresponding sequence of equivalence granules satisfies the condition:

$$[x]_{E_1} \subseteq [x]_{E_2} \subseteq \ldots \subseteq [x]_{E_m}. \tag{4}$$

The nested sequence of equivalence relations produces a multi-level partitions of the universe. This leads to a simple multi-level granulation structure of the universe. Different granulations of the universe form a linear order. A partition is either a refinement or a coarsening of another, although some granules in different levels may be the same.

Let $U = \{a, b, c, d, e, f\}$ be a universe. An example of hierarchical granulation defined by a nested sequence of equivalence relations is given by:

$$\begin{aligned}
\pi_5 :&\ \{\{a, b, c, d, e, f\}\},\\
\pi_4 :&\ \{\{a, b, c, d\}, \{e, f\}\},\\
\pi_3 :&\ \{\{a, b\}, \{c, d\}, \{e, f\}\},\\
\pi_2 :&\ \{\{a\}, \{b\}, \{c, d\}, \{e, f\}\},\\
\pi_1 :&\ \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}\}.
\end{aligned}$$

The linear order on the partitions is given by $\pi_1 \preceq \pi_2 \preceq \pi_3 \preceq \pi_4 \preceq \pi_5$. Partition π_5 is the coarsest partition, and π_1 is the finest partition. An

equivalence class in a coarser partition is further divided into smaller equivalence classes in a finer partition.

Another type of multi-level granulation structures can be defined by a nested sequence of coverings or a nested sequence of reflexive binary relations. Each level is a covering of the universe. For two reflexive binary relations R_1 and R_2, U/R_1 is called a finer covering than U/R_2, written as $U/R_1 \preceq U/R_2$, if $R_1 \subseteq R_2$. For a pair of arbitrary coverings τ_1 and τ_2, we say that τ_1 is a finer covering than τ_2, written as $\tau_1 \preceq \tau_2$, if every class in τ_1 is a subset of certain class in τ_2. Suppose we have a sequence of reflexive binary relations:

$$R_1 \subseteq R_2 \subseteq \ldots \subseteq R_m. \tag{5}$$

Each relation defines a covering of the universe U/R_i and together they define multi-level coverings.

An example of multi-level coverings is given by:

$$\begin{array}{ll}
\pi_5: & \{\{a,b,c,d,e,f\}\},\\
\tau_4: & \{\{a,b,c,d\},\{a,b,c,f\},\{a,f\}\},\\
\tau_3: & \{\{a,b,c\},\{a,d,e\},\{a,e,f\},\{a,f\},\{b,c\}\},\\
\tau_2: & \{\{a,b\},\{a,c\},\{a,d\},\{b\},\{e,f\},\{f\}\},\\
\tau_1: & \{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\}\}.
\end{array}$$

A class in a coarser covering is obtained by adding some elements to a class in a finer covering. In other words, more elements are considered to be similar in a coarser covering.

In a hierarchy, one typically associates a name with a cluster such that elements of the cluster are instances of the named category or concept [16]. Partitions (coverings) in higher levels may be viewed as generalization of partitions (coverings) in lower levels, while partitions (coverings) in lower levels as specialization of partitions (coverings) in higher levels. A name given to a cluster in a higher level is more general than a name given to a cluster in a lower level, while the latter is more specific than the former.

3 Information Retrieval

The discipline of information retrieval concerns the organization, analysis, storage, searching, acquisition and dissemination of information. An information retrieval system is designed with the objective of identifying or ranking useful items from a large quantity of stored information in response

to a user request. While information retrieval systems were traditional used in libraries, their most modern use is the Web search engines. Such systems are used extensively by scientists as an effective tool to find relevant information.

3.1 Basic Issues and Problems

Three fundamental issues in information retrieval are document representation, query formulation, and retrieval functions [3, 7, 39, 45, 50]. One needs to design and implement an appropriate scheme to represent the contents of documents, a language to express user queries, and a retrieval function to search for relevant documents. Index terms play the connecting role between documents and queries (users). A document is considered to be relevant to a query if the user submitting the query judges the document to be useful.

Different retrieval models have been developed, such as the Boolean, vector space, and probabilistic models [39, 45]. Although they provide us formal and elegant formulations of information retrieval problem, they suffer from several shortcomings. The classical retrieval models are over-simplification of the real world retrieval problem.

Each IR model represents a document simply as a list of (weighted) terms that appear in the document. The list is typically the results of some statistical analysis of document text. It should be realized that the effectiveness of statistical analysis, although providing us useful information, is limited [45]. To avoid this difficulty, one needs to consider the structure information and semantic information of the document. Two levels of structure information can be used. The document level structure information shows the connection between components of a document. Such information is now readily available from the online searchable and structured documents prepared using some markup languages. The collection level structure information shows the connections of documents. Citation and co-citation analysis and subject clustering of documents are examples that explore the collection level structure information. Natural language analysis tools and ontology can also be used to discover both levels of structure information. In many information retrieval systems, all documents are described in the same level of details. The same document representatives are used independent of individual users. The lack of consideration of the diversity, background and intentions of users effects the performance of IR systems. It is expected that multiple and personalized representations of documents may be more effective.

Similar observations can also be made regarding the issue of query formulation. Typically, a query language is used to express user information needs. A query language may be either too restrictive to be very effective, or too complicated to be practically useful. The problem is made worse when a user is not clear what is being searched for. Many search engines accept natural language queries. However, such queries are in many cases translated into a list of keywords or some simple Boolean expression. An effective information retrieval system should support many query languages and tools for expressing user information needs.

Many information retrieval systems use a simple and single retrieval method independent of users. In addition, retrieval is based on keyword level matching. Documents containing the keywords appearing in the query are retrieved or ranked higher. Other information that may suggest the relevance of documents is not fully explored. Recent studies on text mining, user behavior analysis, and agent technology may provide potential solutions to such a problem [57]. One needs also to explore the potential of multi-strategy retrieval exemplified by meta-search engines.

In summary, on the one hand, many shortcomings of traditional information retrieval systems and Web search engines make them inadequate to support scientific research using the Web. One the other hand, the advances made in related fields, such as text mining, intelligent agents, and markup languages open new doors to expand information retrieval systems. In fact, many different special purpose systems have been implemented. Two examples of such systems are DBLP (http://dblp.uni-trier.de/) and ResearchIndex (http://citeseer.nj.nec.com/cs). The emergence of such systems strongly suggest the needs for a more general framework for the study of information retrieval to support scientific research, especially on the new platform of the Web.

3.2 Document Space Granulations

Document clustering is a widely used technique in information retrieval to reduce computational costs and improve retrieval effectiveness [34, 49]. Documents may be clustered in several ways, such as content based, query based, and citation based approaches. The most commonly way of clustering is content or topic based. Documents with similar content or topic are put into the same cluster [39, 45]. The recent extensive studies and renewed interest on text categorization further explore document clustering [17, 52]. Another type of document clustering methods is the query oriented document clus-

tering. Documents are clustered based on their joint relevance to a set of queries. That is, documents are put into the same cluster if they tend to be relevant at the same to some queries [38, 60]. A similar idea is to use citation and co-citation information for document clustering [11]. Such clustering methods are used in ResearchIndex, in which, for example, co-cited documents are put into a cluster. Other methods for clustering documents is based on special characteristics of documents. For example, documents can be clustered based on authors, journals or conference, as is done in DBLP. Those clustering methods are not only valuable for content oriented retrieval, but also suitable for other special purpose retrieval.

In content based document clustering, a collection of documents is divided into clusters such that each cluster consists of similar documents. A center called centroid is constructed for each cluster to represent all the documents in that cluster [39]. A clustering of documents provide a granulated view of the document collection. One may use either a partition or a covering of the document collection for clustering. A hierarchical clustering of documents is produced by decomposing large clusters into smaller ones. The large clusters offer a rough representation of the document. The representation becomes more precise as one moves towards the smaller clusters. A document is described by different representations at various levels. Hence, a cluster-based IR system implicitly employs multi-representation of documents. Cluster based retrieval is done by comparing a query with the centers of the larger clusters. If the center of the current cluster is sufficiently close to the query, then the query will be compared against the centroid of the smaller clusters at a lower level. In other words, if it is concluded that a document is not likely to be useful using a rough description, then the document will not be further examined using more precise descriptions. Different retrieval methods may also be employed at different levels.

Document clustering only reduces the dimensionality of the document collection while the dimensionality of index terms remains the same. That is, the same number of terms is used for the representation of cluster centers regardless of the level in the document hierarchy. On the other hand, text categorization uses some predefined categories to label or name a cluster, and thus introduces different representations of the same document.

3.3 Query (User) Space Granulations

Like the granulation of document space, one can construct granulated views of query (user) space in several ways, such as content based, document based

approaches [46, 47]. Content based query clustering is similar to content based document clustering [38]. The similarity of queries is evaluated based on index terms used by the queries. Similar queries are grouped together to represent the needs of a group of users. Content based approaches can be easily extended to cluster users based on user profiles or user logs. On the other hand, document based query clustering methods use the overlap of relevant documents, retrieval results, of queries [10, 33]. Although two queries may not be similar according to their contents, they are still considered to be similar due to a large overlap of relevant documents.

Some recent ideas for query clustering are related to the document based query clustering [47]. They are useful in question answering systems or search engines, such as AskJeeves (http://www.askjeeves.com/). Beeferman and Berger [4] suggested to cluster queries by using click-through data, which is implicit relevance information provided by users. Wen et al. [46, 47] combined both content based and document based (through user document clicks) approaches for query clustering.

3.4 A Unified Probabilistic Model

The application of granulated views of both document space and query space is well illustrated in a unified probabilistic model proposed and studied by Robertson et al. [35].

In probabilistic information retrieval models, the relevance of documents to queries is modeled in probabilistic terms. There are four sub-models in the unified model. Maron and Kuhns' model (Model 1) is based on the granulation of query (user) space [27]. Queries are clustered in order to compute the probability of relevance of a document to a group of queries (users). In contrast, Robertson and Sparck Jones' model (Model 2) is based on the granulation of document space [36]. Documents are clustered in order to compute a probability of relevance of a group documents to a given query (user). With respect to the granulation of both document document and query spaces, one obtains a lower model (Model 0) which computes the probability of relevance of a group of documents to a group of queries (users). When both document and query spaces are not granulated, a high level model (Model 3) is obtained. The high level model represents the ideal situation where the relevance of individual documents to individual queries is used. However, it is difficult to work with Model 3 in practice, as we always have granulated views of either document or query space. Any document representation scheme or query language leads to granulated view

of document or query space. Documents (queries) with the same description are considered to be a whole unit and can not be differentiated. A solution to this difficulty is the combination of both Model 1 and Model 2 [35]. More specifically, the relevance of a particular document to a particular query is estimated by the relevance of the document to a group of queries and the relevance of a group of documents to the query.

3.5 Term Space Granulations

The problem of term clustering and its application in information retrieval have been studied by many authors [5, 32, 37, 39, 40, 42]. In a term hierarchy, a cluster may be assigned new terms as labels of the cluster. The new terms are more general than each individual term in the cluster. In general, a multi-level coverings may be more suitable. A more specific term may be described by more than one general terms.

A term hierarchy serves as an effective tool to summarize knowledge about a specific domain. Many domain-specific term hierarchies or concept hierarchies have been used in the organization and retrieval of scientific literature. Examples of term hierarchies are the ACM Classification System (http://www.acm.org/class/) and the Mathematics Subject Classification (http://www.ams.org/msc/). With such systems, one immediately derives a hierarchical granulation of documents. At each level, documents are described by different terms of different specificities. Documents described by the same terms in the classification system are naturally put into the same cluster.

A main consideration in using term hierarchies is the trade-off relationship between the high dimensionality of index terms and the accuracy of document representation. One may expect a more accurate document representation by using more and specific index terms. However, the increase of the dimensionality of index terms also leads to a higher computational cost.

Term clustering techniques have been used mainly for retrieval, such as query modification, query expansion, and sophisticated retrieval functions [39]. Their use in the granulation of document space and query space has not been fully investigated [51]. The potential of a term (concept) hierarchy, especially its implied knowledge structure, needs to be fully exploited. The fast growing interest on ontology clearly demonstrates a trend in exploring relationship between terms (concepts).

3.6 Retrieval Results Granulations

In many situations, the list of documents returned by information retrieval systems such as search engines is too long and contains duplicate or near-duplicate documents [18]. In order to resolve those problems, many authors suggested and studied the granulation of retrieval results [1, 13, 19, 6, 43, 62]. By clustering retrieval results, one can organize the results and provide coarse-grained summarization to users.

The idea of granulating retrieval results was first studied by Preece [31, 43]. Willet [48] referred to such document clustering as query specific document classification, in contrast to the query independent document granulation discussed earlier. With respect to a particular query, clustering results are more effective [6, 13, 43].

An important issue in query specific document clustering is to obtain a meaningful description of the derived clusters to be presented to the user [19, 6]. The classical centroid based approach is no longer appropriate. It has been suggested that a few titles and some terms can be used as the description of a cluster [6, 13, 19]. One may also extract some important sentences from the documents in a cluster as a description of the cluster.

3.7 Structured and XML Documents

The granulated views of documents and terms can be exploited to discover collection level structures and organize documents accordingly. The document level structure information can be obtained from the use of markup language.

In information retrieval, full text and structured documents have been considered by many authors [2, 14]. With the development of XML (eXtensible Markup Language, http://www.w3.org/TR/REC-xml), there is a growing interest in the organization and retrieval of structured and semi-structured documents. XML is becoming a new standard for data representation and exchange. More XML documents are expected to be available on the Web.

In XML, the structures and the meaning of data are explicitly indicated by element tags. The structure of a document and element tags are defined through a DTD (Document Type Definitions). For example, a scientific article has a hierarchical (multi-level) granulated description given by the tree structure of DTD. An XML document can be viewed in many different ways by focusing on different tags. For example, one can easily extract only the-

orems from an XML document. More specifically, an XML document itself can be viewed as the physical view of the documents, and many different logical views can be obtained.

We can explore the rich information provided in XML documents. For example, one can cluster documents using certain tag fields. In this way, different granulated views of the collection can be formed with respect to different users. An XML document collection also allows multi-strategy retrieval. One may use structured queries by focusing on certain tags or perform free text retrieval by simply ignoring all tags. The rich information available in XML documents enable us to extend the traditional functionalities of information retrieval systems.

4 Evolution of Retrieval Systems

In order to extend information retrieval systems and search engines, we need to move beyond the simple retrieval and search centered design philosophy. To search for a new design philosophy of IRSS, we examine the evolution process of retrieval systems and the roles played by each type of retrieval systems [56]. It is concluded that IRSS is the next evolutionary stage of retrieval systems, which deals with more difficult and more complex problems, and provides more supporting functionalities to a user.

4.1 From Data Retrieval Systems (DRS) to Information Retrieval Systems (IRS)

In one of the classic books on information retrieval (IR), van Rijsbergen compared data retrieval (DR) and IR to illustrate the range of complexity of each mode of retrieval, as well as their common features and differences [45]. DRS may be considered as an early stage, and IRS as the next evolutionary stage in the development of retrieval systems.

In both modes of retrieval, on the one hand, there is a set of information items (data or documents), and on the other hand, there is a user information need. The function of a retrieval systems is to match items with the user need. Both DR and IR focus on the retrieval functionality, namely, the match of items and user information needs. It is not surprising that many results from the discipline of pattern recognition, which deals with pattern match, are applied into IR [9].

The differences between DR and IR can be seen from the ways in which information items and user information needs are represented, as well as

the matching process [3, 45, 50]. A database system is a typical example of DR. In DR, data items and user information needs can be precisely described by using well understood knowledge representation schemes and query languages. The model is deterministic in the sense that the relationships between data items and user needs are well and objectively defined, and consequently exact match and deductive inference can be used. In contrast, in IR, documents and user needs can not be precisely described, and their relationships are ill and subjectively defined. The model is non-deterministic in the sense that partial or best match and inductive inference are used. In summary, DR deals with structured, and well-defined problems in which there is no uncertainty, while IR deals with semi-structured, unstructured, and ill-defined problems where uncertainty plays a major role.

In the design of IRS, the search functionality of DRS is extended. However, the design philosophy remains to be the same. Both DRS and IRS are designed to provide essentially the search functionality.

Some of the reasons for adopting the search functionality centered design philosophy can be seen from the historical roles played by IR systems. IR systems were first introduced in libraries for retrieval information [3]. Baeza-Yates and Ribeiro-Neto divide IR systems into three generations [3]. The first generation is basically an automation of conventional manual catalog search, which allows searches using author name and title. The second generation increases the search functionalities by allowing text based search, such as searching by subject headings, keywords, and more complex queries. The third generation focuses on improved graphical interfaces, hypertext features, and open system architectures. We can conclude that IR systems, in the context of libraries, only attempt to automate the task of search, while other user support functionalities are left to the librarians.

Baeza-Yates and Ribeiro-Neto differentiate two distinct types of user tasks when using a retrieval system, the retrieval (search) task and the browsing task [3]. A retrieval task is normally performed by translating an information need into a query and searching using the query. A browsing task is carried out by looking around in a collection of documents through an interactive interface. During browsing the user information need or objective may not be clearly defined, and can be revised through the interaction with the system. The third generation of IR systems provide more functionalities for browsing. As summarized by Baeza-Yates and Ribeiro-Neto, "Classic information retrieval systems normally allows information or data retrieval. Hypertext systems are usually tuned for providing quick browsing. Modern digital library and Web interfaces might attempt to combine these tasks to

provide improved retrieval capabilities."

A careful examination of the three generations of IR systems shows that each later generation provides enhanced or more functionalities of the previous generation. As we move from DR to IR, as well as from one generation of IR systems to another, two dimensions of changes can be observed. One dimension concerns the complexity and the nature of the problems, ranging from structured, semi-structured, to unstructured. DR deals with structured problems, where all involved concepts can be precisely defined. IR deals with semi-structured problem, where some concepts can not be precisely defined. Future systems may deal with unstructured problems. Another dimension concerns the user control or user tasks, ranging from simple to complex. DR systems deal with fact retrieval, IR systems deal with non-fact searching and browsing. Future generations of systems may deal with more complex user tasks, such as analysis, organization, and discovery. This two-dimensional description is adopted from a well known framework for decision support systems (DSS) [44].

4.2 From Information Retrieval Systems (IRS) to Information Retrieval Support Systems (IRSS)

Many new developments in IR have been made, ranging from multimedia (images, audio or video) retrieval, hypertext retrieval, and digital library to Web information retrieval [3]. With the rapid development of the Web and digital libraries, we have witnessed a wider range of applications of IR systems, and a renewed interest in IR. In fact, IR systems such as search engines play an important role for the success and boom of the Web. On the other hand, the design philosophy and principles of IR, well discussed in classic textbooks [39, 45], have remained more or the less the same. The observation by Lesk that "the more things change, the more they stay the same" is still applicable today as it was applicable 10 years ago [24]. To a large extent, information retrieval can still be viewed as document retrieval by substituting 'document' for 'information', as pointed out by van Rijsbergen long time ago [45]. The fundamental ideas of IR systems, namely, indexing and searching, have remained to be the same. IRS are perceived as systems that provide the basic search and browsing functionalities.

From the two-dimensional description of DR and IR problems, we need to consider unstructured problem and more user control in the design and implementation of new information retrieval strategies and systems. We need to have a better understanding of user tasks. The ultimate goal of

finding information is to use the relevant information, say, in a decision making process. For example, a researcher may compare, analyze and summarize the relevant information in preparing a scientific article, or evaluating a project proposal. In order to find and extract useful information from a large document collection or the Web, as well as effectively use the extracted information in problem solving, a user must play an active role in various tasks, such as browsing, investigating, analyzing, understanding, organizing, and searching the collection. Searching and browsing are only some of the simple and front end tasks. The next generation of IR systems must support more types of user tasks, in addition to searching and browsing. Since these tasks can not necessarily be described precisely, fully automation can not be expected. Instead, one can build various tools, methodologies, and languages for supporting such user tasks.

From the above discussion, we can conclude that the search centered philosophy for the design of IR system may no longer be suitable. This is also evident from the fact that many Web search engine users must spend more time to understand, filter, and organize documents returned by a search engine. With the Web as a new media for information storage, delivery, gathering, sharing, processing, and utilization, the problem of information retrieval is no longer a simple process of search.

A new set of philosophy and principles for the design and implementation of the next generation IR systems is needed. Instead of focusing on the search functionality, one focuses on the supporting of functionality [58]. This can be viewed as the next stage in the evolution of retrieval systems, which leads to the introduction of information retrieval support systems (IRSS). The objective of an IRSS is to support many different types of user tasks in finding and utilizing information, in a similar way that a decision support system (DSS) assists users in making managemental decisions [44].

5 Basic Issues of IRSS

In this study, the term information retrieval is used as a much broader umbrella term to cover any and every user tasks in finding and utilizing information from a collection of documents or the Web when performing scientific research. Similarly, the term IRSS is used as an umbrella term to describe any and every computerized systems used to support and improve information retrieval. In a more accurate sense, we perhaps should use the term Research Support System (RSS). We choose the term IRSS to

emphasize the particular aspects of research support covered by conventional IR systems.

5.1 The Concept of IRSS

Our interpretation of IRSS draws extensively results from the related field of decision support systems (DSS) [44]. While DSS focus on supporting and improving decision making, IRSS focus on supporting and improving retrieval. The philosophy, principles, and techniques from DSS are applicable to IRSS by simply substituting the tasks of "decision making" for the tasks of "information retrieval". This view of IRSS is particular reasonable, if one considers the fact that it is necessary to obtain useful information in order to make intelligent and rational decision.

A classical definition of DSS given by Gorry and Morton [12] defines DSS as "interactive computer-based systems, which help decision makers utilize data and models to solve unstructured problems." Turban and Aronson compare different definitions of DSS and suggest the following one [44]:

> "Computer-based information systems that combine models and data in an attempt to solve nonstructured problems with extensive user involvement through a friendly user interface."

From the above definition, we want to stress two important features of DSS pertinent to our interpretation of IRSS. One feature is the combination of data and models. Data are raw and uninterpreted facts. In order to make sense of raw data, one needs to construct various models. Therefore, an DSS deals with both data and their interpretations. The other feature is the emphasis on the user involvement. An DSS plays a supporting role in problem solving.

One can give a formal definition of IRSS based on the definition of DSS. However, the definition of IRSS is not the focus of this chapter. It is perhaps wise to leave the notion loosely defined until we have gained more insights into the problem. For the time being, we rely on our intuitive understanding of IRSS.

5.2 Characteristics of IRSS

The problems of finding and using relevant information from a large collection of documents are unstructured problems that can not be easily and precisely described. It is made even more complicated by the fact that a user

may not know exactly what is being searched for. In solving the retrieval problems, IRSS are more useful and effective than IRS. Based on the two features of DSS mentioned earlier, we can identify some characteristics that distinguish IRSS from current IRS, and show the advantage of IRSS.

Current IRS emphasis on the storage and search functionalities, which leads to a lack of consideration of the two important issues, namely, models and user involvement. In other words, an IRS performs search at the raw data level, instead of the model level, and without user interaction. Although recent IRS systems exemplified by Web search engines build hierarchical model to provide semantic interpretation of documents in a collection, the end users are not involved in the model construction process. A remedy to this problem is the use of personalized user profile and personalized bookmarks.

IRSS attempt to resolve the problems of IRS by providing more supporting functionalities. An IRSS provides models, languages, utilities, and tools to assist a user in investigating, analyzing, understanding, and organizing a document collection and search results. These tools allow the user to explore both semantic and structural information of each individual document, as well as the entire collection.

Three related types of models need to be considered in IRSS. Documents in a document collection serve as the raw data of IRSS. The document models deal with representations and interpretations of documents and the document collection. The retrieval models deals with the search. The presentation models deal with the representation and interpretations of results from the search. A single document model, a retrieval model, or presentation model may not be suitable for different types of users. Therefore, IRSS must support multi-model, and provide tools for users to manage various models.

The document models allow multi-representation of documents. Granular computing will play an important role in the construction of document models [51, 58, 61]. One can use a hierarchical granulation of document collection, namely, a layered and multi-resolution representation of documents. The same document is represented in less detail at a higher level than at a lower level. With the multi-model capability, a user can create different logical views of a document collection and logical links between documents, as well as compare and investigate various views. The resulting in-depth knowledge can help the user to locate and infer useful information. The recent development of XML enables us to describe both the structural and semantic information of a document. Such information makes the construc-

tion of multi-document models an achievable goal.

The retrieval models provide languages and tools to assist a user to performs tasks such as searching and browsing. IRSS should provide multi-strategy retrieval. A user can choose different retrieval models with respect to different document models. The presentation models allow a user to view and arrange search results, as well as various document models. The same results can be viewed in different ways by using distinct presentation models. Moreover, a user can analyze and compare results from different retrieval models.

IRSS implement and manage three types of models, as well as the associated languages, tools and utilities. An IRSS is highly interactive so that a user can make decisions at various stages. A user plays a more active role in the process of finding useful information. There are many advantages for the extensive user involvement. The user involvement is particularly important in exploratory type of searching and browsing. The usefulness or the relevance of each information item (i.e., a group of documents, a document, and parts of a document) can only be determined by the user. Without the involvement of a user in the analysis and organization of a document collection, one can not expect the user to provide a meaningful query.

From the previous discussion, we can conclude that an IRS performs easy and loosely structured retrieval tasks, where automation is possible and user involvement is not necessary. On the other hand, an IRSS supports a user to perform difficult and unstructured retrieval problems. Since a fully automation is impossible, at least not for the time being, the main function of IRSS is to support a user. The retrieval process is controlled by the user.

5.3 Components of IRSS

One may argue about the exact components of an IRSS. We take a simple approach by adopting the results from DSS and intelligent systems.

According to Turban and Aronson, a DSS normally consists of four subsystems: [44].

- *Data management subsystem*: This subsystem deals with lower level raw data management using software systems such as database management system (DBMS) and data warehouse.

- *Model management subsystem*: This subsystem is referred to as a model base management system (MBMS). It includes existing quanti-

tative models for analyzing and interpreting the raw data, and provides language and tools for building user models.

- *Knowledge-based management subsystem*: This subsystem supports other subsystem and provides intelligence to a decision maker.
- *User interface subsystem*: This subsystem handles the interaction between user and the system.

The above schematic description of DSS can be applied to the study of IRSS. In other words, although the objects managed by each subsystem may be different, the fundamental principles are the same.

5.4 Fields Related to IRSS

Techniques, results and lessons from many fields can be used in the study of IRSS and to enhance the capabilities of an IRSS. A few related fields are summarized below, in addition to the previously discussed DSS and GrC.

Expert systems (ES).

A well established practice in expert systems is the separation of knowledge and inference engine. While the inference engine is logic based and problem independent, the knowledge base is domain specific. Expert system shells that implement inference engine can be used to build many different domain-specific expert systems. Similarly, we can build knowledge based IRSS by separating knowledge base and management subsystem. An IRSS shell can be built that provide a set of domain and user independent tools, using which domain specific IRSS can be constructed.

Users of retrieval systems may fall into many different categories, have different background, and with different types of information needs. Documents in a collection may also cover different domains. One can not expect to design a system that is best for everyone and for every domain. Consequently, one needs to study principles, methodologies, and techniques that can be used to design and implement domain-specific IRSS.

Another feature of expert systems is the explanation functionality. An expert system not only provides a solution, but can also explain why and how the solution is arrived. It is reasonable to insist on the explanation functionality of an IRSS.

Machine learning, data mining and text mining.

By applying algorithms of machine learning, data mining, and text mining to documents stored in an IRSS, one may discover patterns and extract

knowledge useful to a user. Such functionalities are particularly useful to users who are interested in exploratory searching and browsing. For instance, a user can track trends in a particular area or discover emerging topics from the constantly changing document collection. A user may also discover links between different documents or research areas.

Computer graphics and data visualization.

In many cases, a user may not want details about particular documents that contain the useful information. A user may want to have a general feeling before going to a more in-depth analysis. With the granulation of document collection, it is possible to provide a user with granulated view, in which details are omitted. Most current IRS present search results in the form of ranked list of individual documents. In an IRSS, a user should be able to use graphics and visualization tools to view a particular document model. Visualization enables a user to perform high level inference and analysis.

Intelligent information agents.

Intelligent information agents have been used by many IRS to collect information and interact with users. The potentials of agents need to be further explored in IRSS. In particular, a user should be allowed to construct a personalized agent to interact with an IRSS. The autonomy and learning capabilities of agents make them attractive to both IRSS and users.

In summary, an IRSS can incorporate any particular type of information systems to provide a specific type of support.

6 A Granular Computing Model for Organizing and Retrieval XML Documents

In this section, we provide a concrete example to illustrate the ideas developed in the previous sections. We will focused mainly on the granulation of an XML document collections.

For simplicity, we consider scientific XML documents given by:

```
<Paper>
   <Title>  </Title>
   <Author>
      <Name>
        <First>  </First>
        <Last>   </Last>
```

```
      </Name>
      <Institution> </Institution>
   </Author>
   <Abstract>
      <Paragraph>  </Paragraph>
      ...
    </Abstract>
    <Section>
        <Title>  </Title>
        <Subsection>
          <Title>  </Title>
          <Paragraph>  </Paragraph>
          ...
        </Subsection>
        ...
    </Section>
    ...
</Paper>
```

Many tags, such as definition, theorem, etc., are not included, as the simple layout is sufficient for the purpose of illustration. It follows that the structure of an article can be clearly observed.

We have a natural multi-level granulation of a scientific document. The granulated representation of an article is the document level granulation. The semantic information provided by tags allows us build different logical views of the same document. For example, one can read only the section titles or the abstract to gain general knowledge about the article. One may also zoom in to particular paragraphs, definitions, or theorems for more details.

At higher levels of granulation, namely, the collection level granulation, we seek for relationships between individual XML documents. Such relationships can be derived from information in each document. In fact, a tag or a combination of tags may be used to cluster documents together to form granules. For example, the collection may be divided by the author name and each granule consists of articles published by that author. Each such granule may be further divided by the year of publication, so that we can obtain a more detailed picture of the publication record of that author. Similarly, documents can be grouped together by journal names, and further divided by the year of publication. Documents may be grouped together by topics,

as represented by index terms (keywords) in the title, section titles, or the entire text, or citation. Those logical views of collection have also been used in some digital library systems, such as DBLP (http://dblp.uni-trier.de/) and ResearchIndex (http://citeseer.nj.nec.com/cs). However, each system has its predefined logical views, and a user is not allowed to create his/her own logical view. By providing the user with more flexibility, one can build a more flexible information retrieval support system.

With respect to the classification of document by keywords, we can build a hierarchical granulation structure. Since the title of an article is the coarsest representation, we can obtain the first level of granulation by using keywords in the title. It is expected that large sized granules would be produced. Each granule can in turn be further divided by keywords in the titles of sections and subsections. The resulting granules can be further divided by using keywords in the full text of the article. The article level granulation forms the remaining part of the hierarchy. At each level, the same document is represented differently. For example, at the collection level, it is represented as different lists of keywords, while at the article level, it is represented in XML format. The granulation of the collection enables the user to understand structural information about the collection. Such useful information may not be obtained from fixed and pre-determined logical views provided by some systems. In searching for useful information, the user can control the level of details through interaction with a retrieval support system. In the process of granulation, the user can enquiry the information of the documents in particular granules. Different retrieval strategies and methods may be employed. For example, at a higher level of the hierarchy, the user may choose a Boolean model to eliminate apparently uninteresting documents. At a lower level, the user may use the vector space model to obtain more accurate and ranked results.

In the GrC model, the three basic types of operations, creation of logical views (granulation), navigation through different logical views, and retrieval, support the user to investigate, to analyze, to understand the collection, and to search for useful information from the collection.

7 Conclusion

In this chapter, we discuss two related topics, namely, the application of granular computing to information retrieval and the introduction of Information Retrieval Support Systems (IRSS).

It is demonstrated that granulation plays an important role in information retrieval. Many entities in information retrieval can be granulated to obtain coarse-grained views. One can granulate document space, query (user) space, term space and retrieval results. The granulated views allow us to focus on the useful structures without looking into too much details. Many existing studies deal with the granulation of a specific entity. The interactions between granulations of different entities need further investigation.

Our formulation and understanding of IRSS draw extensively results from DSS, which in turn draw results from many fields. It represents a new design philosophy to information retrieval. It is hope that more attention can be drawn to this important topic. IRSS may be viewed as the next stage in the evolution of retrieval systems. IRSS focus on various supporting functionalities of retrieval systems, in addition to support search and browsing. IRSS also stress the importance of extensive user involvement.

The new philosophy of retrieval support, rather than retrieval, may fundamentally change the design goal of current IR systems, which enables us to move from simple retrieval systems to advanced retrieval support systems.

References

[1] R.B. Allen, P. Obry and M. Littman, An interface for navigating clustered document sets returned by queries, *Proceedings of the ACM Conference on Organizational Computing Systems* (1993) pp. 166-171.

[2] J. André, R. Furuta and V. Quint (eds.), *Structured Documents* (Cambridge, Cambridge University Press, 1989).

[3] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval* (New York, Addison Wesley, 1999).

[4] D. Beeferman and A. Berger, Agglomerative clustering of a search engine query log, *Proceedings of KDD'00* (2000) 407-415.

[5] H. Chen, T. Ng, J. Martinez and B. Schatz, A concept space approach to addressing the vocabulary problem in scientific information retrieval: an experiment on the worm community system, *Journal of the American Society for Information Science* Vol. 48 (1997) pp. 17-31.

[6] C. de Loupy, P. Bellot, M. El-Bèze and P.F. Marteau, Query expansion and classification of retrieved documents, *Proceedings of the Seventh Text REtrieval Conference (TREC-7)* (1998) pp. 382-389.

[7] S. Dominich, *Mathematical Foundations of Information Retrieval* (Dordrecht, Kluwer Academic Publishers, 2001).

[8] M.P. Courtois, and M.W. Berry, Results ranking in Web search engines, *Online* Vol. 23 (1999) pp. 39-46.

[9] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis* (New York, Wiley, 1973).

[10] L. Fitzpatrick and M. Dent, Automatic feedback using past queries: social searching? *Proceedings of SIGIR'97* (1997) pp. 306-313.

[11] Garfield, E. *Citation Indexing – Its Theory and Application in Science, Technology and Humanities* (New York, John Wiley & Sons, 1979).

[12] G.A. Gorry and M.S.S. Morton, A framework for management information systems, *Soloan Management Review* Vol. 13 (1971) pp. 55-70.

[13] M.A. Hearst, J.O. Pedersen, Reexamining the cluster hypothesis: Scatter/Gather on retrieval results, *Proceedings of SIGIR'96* (1996) pp. 76-84

[14] F.C. Heeman, Granularity in structured documents, *Electronic Publishing* Vol. 5 (1992) pp. 143-155.

[15] J. Heflin, R. Volz, and J. Dale (eds.), *Requirements for a Web Ontology Language*, W3C Working Draft, 07 March 2002, http://www.w3.org/TR/webont-req/

[16] N. Jardine and R. Sibson, *Mathematical Taxonomy* (New York, Wiley, 1971).

[17] T. Joachims, Text categorization with support vector machines: learning with many relevant features, *Proceedings of the 10th European Conference on Machine Learning* (1998) pp. 137-142.

[18] J.W. Kirriemuir and P. Willet, Identification of duplicate and near-duplicate full-text records in database search-outputs using hierarchic cluster analysis, *Program* Vol. 29 (1995) pp. 241-256.

[19] Y. Kural, S. Robertson and S. Jones, Clustering information retrieval search outputs, *21st Annual BCS-IRSG Colloquium on IR* (1999), http://www1.bcs.org.uk/DocsRepository/02800/2856/kural.pdf

[20] S. Lawrence, Online or invisible, *Nature* Vol. 411 (2001) 521.

[21] S. Lawrence and G.L. Giles, Searching the World Wide Web, *Science* Vol. 280 (1998) pp. 98-100.

[22] S. Lawrence and G.L. Giles, Context and page analysis for improved Web search, *IEEE Internet Computing* Vol. 2 (1998) pp. 38-46.

[23] S. Lawrence, G.L. Giles and K. Bollacker, Digital libraries and autonomous citation indexing, *IEEE Computer* Vol. 32 (1999) pp. 67-71.

[24] M. Lesk, SIGIR'91: the more things change, the more they stay the same, *SIGIR Forum* Vol. 25 (1991) pp. 4-7.

[25] T.Y. Lin, Y.Y. Yao, and L.A. Zadeh (eds.), *Data Mining, Rough Sets and Granular Computing* (Heidelberg, Physica-Verlag, 2002).

[26] W. Marek and H. Rasiowa, Gradual approximating sets by means of equivalence relations, *Bulletin of Polish Academy of Sciences, Mathematics* Vol. 35 (1987) pp. 233-238.

[27] M.E. Maron and J.L. Kuhns, On relevance, probabilistic indexing and information retrieval, *Journal of the Association for Computing Machinery* Vol. 7 (1960) pp. 216-244.

[28] A. Odlyzko, The rapid evolution of scholarly communication, http://www.si.umich.edu/PEAK-2000/odlyzko.pdf

[29] Z. Pawlak, *Rough Sets, Theoretical Aspects of Reasoning about Data* (Dordrecht, Kluwer Academic Publishers, 1991).

[30] J.A. Pomykala, A remark on the paper by H. Rasiowa and W. Marek: "Gradual approximating sets by means of equivalence relations", *Bulletin of Polish Academy of Sciences, Mathematics* Vol. 36 (1989) pp. 509-512.

[31] S.E. Preece, Clustering as output option, *Proceedings of the American Society for Information Science* (1973) pp. 189-190.

[32] Y. Qiu and H. Frei, Concept based query expansion, *Proceedings of the Sixteenth ACM International Conference on Research and Development in Information Retrieval* (1993) pp. 160-169.

[33] V.V. Raghavan and H. Sever, On the resuse of past optimal queries, *Proceedings of SIGIR'95* (1995) pp. 344-350.

[34] E. Rasmussen, Clustering algorithms, in W. Frakes and R. Baeza-Yates (eds.) *Information Retrieval: Data Structures and Algorithms* (Englewood Cliffs, Prentice Hall, 1992) pp. 419-442.

[35] S.E. Robertson, M.E. Maron and W.S. Cooper, Probability of relevance: a unification of two competing models for document retrieval, *Information Technology: Research and Development* Vol. 1 (1982) pp. 1-21.

[36] S.E. Robertson and K. Sparck Jones, Relevance weighting of search terms, *Journal of the American Society for Information Science* Vol. 27 (1976) pp. 129-146.

[37] H. Sakai, K. Ohtake and S. Masuyama, A retrieval support system by suggesting terms to a user, *Proceedings 2001 International Conference on Chinese Language Computing* (2001) pp. 77-80.

[38] G. Salton, *Dynamic Information and Library Processing* (Englewood Cliffs, Prentice-Hall, 1975).

[39] G. Salton and M. McGill, *Introduction to Modern Information Retrieval* (New York, McGraw Hill, 1983).

[40] P. Schäuble and D. Knaus, The various roles of information structures, *Proceedings of the Sixteenth Annual Conference of the Gesellschaft für Klassifikation* (1993) pp. 282-290.

[41] R. Slowinski and D. Vanderpooten, A generalized definition of rough approximations based on similarity, *IEEE Transaction of Knowledge and Data Engineering* Vol. 12 (2000) pp. 331-336.

[42] K. Spark Jones, *Automatic Keyword Classification for Information Retrieval* (London, Butterworths, 1971).

[43] A. Tombros, R. Villa and C.J. van Rijsbergen, The effectiveness of query-specific hierarchic clustering in information retrieval, *Information Processing and Management* Vol. 38 (2002) pp. 559-582.

[44] E. Turban and J.E. Aronson, *Decision Support Systems and Intelligent Systems* (New Jersey, Prentice Hall, 2001).

[45] C.J. van Rijsbergen, *Information Retrieval* (London, Butterworths, 1979).

[46] J.R. Wen, J.Y. Nie and H.J. Zhang, Clustering user queries of a search engine, *Proceedings of the Tenth International World Wide Web Conference* (2002) pp. 162-168

[47] J.R. Wen, J.Y. Nie and H.J. Zhang, Query clustering using user logs, *ACM Transactions on Information Systems* Vol. 20 (2002) pp. 59-81.

[48] I. Willett, Query specific automatic document classification, *International Forum on Information and Documentation* Vol. 10 (1985) pp. 28-32.

[49] I. Willett, Recent trends in hierarchic document clustering: a critical review, *Information Processing and Management* Vol. 24 (1988) pp. 577-597.

[50] S.K.M. Wong and Y.Y. Yao, On modeling information retrieval with probabilistic inference, *ACM Transactions on Information Systems* Vol. 13 (1995) pp. 38-68.

[51] S.K.M. Wong, Y.Y. Yao and C.J. Butz, Granular information retrieval, in F. Crestani and G. Pasi (eds.) *Soft Computing in Information Retrieval: Techniques and Applications* (Heidelberg, Physica-Verlag, 2000) pp. 317-331.

[52] Y. Yang and J.O. Pedersen, A comparative study on feature selection in text categorization, *Proceedings the 14th International Conference on Machine Learning* (1997) pp. 412–420.

[53] Y.Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, *Information Sciences* Vol. 111 (1998) pp. 239-259.

[54] Y.Y. Yao, Granular computing: basic issues and possible solutions, *Proceedings of the 5th Joint Conference on Information Sciences* Vol. I (2001) pp. 186-189.

[55] Y.Y. Yao, Information granulation and rough set approximation, *International Journal of Intelligent Systems* Vol. 16 (2001) pp. 87-104.

[56] Y.Y. Yao, Informaiton retrieval support systems, *Proceedings of the 2002 IEEE World Congress on Computational Intelligence* (2002) pp. 773-778.

[57] Y.Y. Yao, H.J. Hamilton and X. Wang, PagePrompter: an intelligent Web agent created using data mining techniques, *Proceedings of International Conference on Rough Sets and Current Trends in Computing*, LNAI 2475 (2002) pp. 506-513.

[58] Y.Y. Yao, K. Song and L.V. Saxton, Granular computing for the organization and retrieval of scientific XML documents, *Proceedings of the Sixth International Conference on Computer Science and Informatics* (2002) pp. 377-381.

[59] Y.Y. Yao and N. Zhong, Granular computing using information tables, in T.Y. Lin, Y.Y. Yao and L.A. Zadeh, (eds.) *Data Mining, Rough Sets and Granular Computing* (Heidelberg, Physica-Verlag, 2002) pp. 102-124.

[60] C.T. Yu, Adaptive document clustering, *Proceedings of SIGIR'85* (1985) pp 197-203.

[61] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems* Vol. 19 (1997) pp. 111-127.

[62] O. Zamir and O. Etzioni, Web document clustering: a feasibility demonstration, *Proceedings of SIGIR'98* (1998) pp. 46-54.