DESIDOC Journal of Library & Information Technology, Vol. 31, No. 3, May 2011, pp. 189-201

Content-Based Document Recommender System for Aerospace Grey Literature: System Design

K. Nageswara Rao and V.G. Talwar*

Defence Research & Development Laboratory, Hyderabad-500 058
E-mail: nageswararao@drdl.drdo.in
**Vice-Chancellor, University of Mysore, Mysore-570 005*

ABSTRACT

Recorded knowledge in the form of manuscripts, print documents, microforms, CD-ROMs, computer files, etc., is increasing exponentially. In order to locate and access relevant information from this vast amount of literature, efforts have been made from time and on to develop various tools and techniques. Early evolved tools/techniques include: library catalogues, indexes, concordances and so on. Information Retrieval Systems played a vital role in the field of information & librarianship to find relevant information from vast number of documents. Recently intelligent agents gained importance as they are able to query databases and resources on Internet, remote library catalogs thereby reducing information overload on the user. Another technology that alleviates the information overload problem is the filtering or recommender systems. The purpose of development of recommender systems is to provide useful and most relevant recommendations or suggestions from number of available alternatives. The present study aimed at design & development of Content-based Document Recommender System (CODORS) to retrieve most relevant technical documents without necessarily matching title terms and closely related to a particular search term(s) as opposed to general Online Public Access Catalog (OPAC) search results. The developed CODORS converts terms expressed by the user in natural language automatically into subject descriptors, carry on search, ranks and retrieves documents.

Keywords: Recommender systems; information retrieval systems, content-based document recommender system, CODORS

1. INTRODUCTION

In day-to-day life, we see that when an individual is not in a position to choose from the available alternatives in a particular situation, he/she goes for the opinion/help of others who have successfully resolved in such a situation. Basically, the criterion followed by the individual to solve his problem is to seek for "*opinion/help of others*". To address these problems more effectively, several recommender systems were developed by the research community depending on the context. From the literature, it is seen that some researchers have used the concepts "*recommender system*", "*collaborative filtering*", and "*social filtering*" interchangeably to connote a system that retrieves and suggests most relevant items based on user profile. Conversely, others regard "*Recommender System*" as a generic descriptor that represents various recommendation/prediction techniques including collaborative, social, and content based filtering, Bayesian networks, association rules, etc. As the name suggests, the task of the recommender system is to suggest items or products to the customer based on his/her preferences. Majority of the these systems are developed and utilized by E-commerce websites as marketing tools to increase revenue by presenting products that the customer is likely to buy and to increase customer loyalty.

There are many difficulties in developing good model of user's interests. However, several factors could be considered to describe user's interest. For example, a user provides set of keywords or terms to describe his interests. Other sources of information such as: which documents the user has read in the past?, what organization the user work in?, which books the user has purchased?, which association the user belongs?, etc., could be used to build user profiles. Even though there is a clear cut idea of what factors are important for

Received 10 January 2010

predicting user interests, there is no guarantee that those factors can decide the information requirements of users. A simple method of determining whether information matches a user's interest is through keyword matching. If user's interests are described by certain keywords, then information containing those words should be relevant. But in reality it is not the case because inappropriate matches arise when people do not exactly reflect the topic or content. It is due to the fact that a single word can have more than one meaning, and conversely, the same concept can be described by many different words. Furnas, Landauer, Gomez, and Dumais[1] have conducted experiments to study the word usage to describe or name objects in various domains so that other people would be able to identify commonly used words. They found that the average likelihood of any two people using the same word in describing the same object ranged from about 0.07 to 0.18 for the different domains.

2. NEED FOR THE PRESENT STUDY

With ever growing technological revolutions and adoption of various technologies, many libraries and information centers have developed bibliographic databases for books, reports, standards, patents, and implemented OPACs for information retrieval. According to Hildreth[2], online catalogs have been classified according to three generations. In the first generation, OPACs were largely known as item finding tools, typically searchable by author, title and accession number and contained relatively short, non-standard bibliographic records. In the second generation, included increased search functionality i.e. access by subject headings, keyword, some Boolean search capability, and ability to browse subject headings. Also offered a choice of display formats (short, medium, long) and improved usability (error messages). Third generation systems included strategy, assistance, free text, controlled vocabulary input, and individualized displays. Further, features incorporated in new systems include improved graphical user interface, support for Z39.50, hyperlinks, and Dublin core metadata standard and incorporation of Java programming[3-6].

So, reasonable solution for identification of technical documents is to develop a bibliographic database and provide a search tool i.e. Online Public Access Catalog (OPAC). A better solution for retrieving most relevant technical documents is to develop an intelligent information retrieval system or recommender system, which builds the user profile, based on the information requirements of the user and recommend or suggest documents that are highly relevant to the individual user. Presently there exists no such recommender system particularly in the field of aerospace. Hence there is a need for a systematically designed recommender system in the field of aerospace to reduce information overload on scientists & scholars and to facilitate them to access information that is most relevant to their requirement. Design and development of such a system leverages the utility of technical documents and reduces the information overload on the user community

The purpose of designing Content-based Document Recommender System is to identify most relevant technical documents particularly from the host of grey literature, that the user is interested. User information requirements are expressed in natural language terms. The system intends to convert these terms expressed in natural language automatically into subject descriptors, carry on search, rank and retrieve documents. The document ranking is automatically done by the system on the basis of weightages calculated based on the occurrences of number of subject descriptors, which are assigned to each title of the document in the collection. The results are sorted on relative relevance ranking and are presented to the user for maximum utilization of technical resources that are otherwise hidden in the database collection. The CODORS system uses operating system, relational database, programming language and Web server for its effective and efficient implementation. The succeeding sections discuss the hardware platform, operating system and such other issues relating to the design and development of CODORS system.

3. HARDWARE AND SOFTWARE PLATFORM

HP DL 380 G4 Server with the following specification has been chosen for the design and development of general OPAC software and CODORS software. Intel Xeon 3.6 GHz Duel Processor with 800 MHz Front Side Bus, 2 MB L2 Cache Memory, 4 GB PC2-3200R 400 MHz DDR2 RAM, and 2 x 72 GB Hot Plug U320 SCSI Hard Disk Drive. The selection of right operating system depends on current needs of the user, interoperability, and support for different hardware platforms and maintainability. Authors have selected Microsoft Windows 2000 Server as operating system for design, development and implementation of CODORS.

4. DATABASE MANAGEMENT SYSTEM

A database management system (DBMS) is a software system that facilitates creation and maintenance of a database(s), and execution of computer programs to perform functions such as storing, retrieving, adding, deleting and modifying data in database(s). There are different types of DBMS, ranging from small systems that run on personal computers to huge systems that run on mainframe systems. The DBMS manages user requests (and requests from other programs) so that users and other programs are free from understanding where the data is physically stored on storage media. While handling user requests, the DBMS ensures the integrity of data and security. Such database management systems include dBase, Paradox, IMS, Sybase, Microsoft SQL Server, IMB DB2, IBM Informix, MySQL, PostgreSQL and Oracle. For designing and developing CODORS, authors have opted Oracle 9i as relational database due to its availability and expertise on its usage.

5. JAVA PROGRAMMING LANGUAGE

Generally, Web applications allow Websites to be dynamic pages rather than static pages. Java Servlet technology provides Web developers with a simple, consistent mechanism for extending the functionality of a Web server and transforming a Website with static text and images into a rich, dynamic and interactive environment. Before the development of Java Servlet technology, Web developers could add interactive functionality to their sites using CGI (Common Gateway Interface) technology. When the user submits the HTML form data to the Web server, the Web server in turn submits the form data to the CGI program. The CGI program generates the output data or HTML page and submits to the Web server for onward transmission to Web client or user. Due to variety of enhancements over plain CGI programs, the Servlets have occupied the role of CGI programs. Java Servlets are more efficient, easier to use, more powerful, more portable and cheaper than traditional CGI and other CGI-like technologies. The most fundamental advantage of Servlets is that they are tightly coupled to the Web server and works within the Web server environment. Java Server Pages (JSP) is a technology for controlling the content or appearance of Web pages through the use of Servlets, small programs that are specified in the Web page and run on the Web server to modify the Web page content before it is transferred to the user who requested it. Sun Microsystems, the developer of Java, also refers to the JSP technology as the Servlet Application Program Interface (API). JSP is comparable to Microsoft's Active Server Pages (ASP) technology. A Java Server Page calls a Java program that is executed by the Web server, an Active Server Page contains a script that is interpreted by a script interpreter (such as VBScript or JScript) before the page is sent to the user. Therefore, the JSP programming language has been used for design and development of CODORS.

6. WEB SERVER (JAKARTA TOMCAT)

Apache Tomcat is a Java based Web application container developed at the Apache Software Foundation (ASF) to run Servlets and Java Server Pages specifications in Web applications. As part of Apache's open source Jakarta project, it has nearly become the industry accepted standard reference implementation for both the Servlets and JSP API. Tomcat is very good because it provides both Java Servlet and Java Server Pages technologies. As a result, Tomcat is good choice for use as a Web server for many applications and also if one wants a free Servlet and JSP engine. The authors have used Tomcat Web Server for access to the interfaces developed in CODORS.

7. TEST DATABASE OF DOCUMENTS

A database of conference papers & their subject descriptors on aerospace and allied subjects was developed for the purpose of designing, developing and testing CODORS. The Dialog OnDisc Aerospace Database served as the source of these conference papers and subject descriptors. DIALOG OnDisc Aerospace Database is a bibliographic and abstract reference of two print publications, viz: International Aerospace Abstracts (IAA) and Scientific and Technical Aerospace Reports (STAR). The subject descriptors in DIALOG OnDisc Aerospace Database are assigned using NASA Thesaurus and as such they represent the thought content of documents indexed in the said database.

8. DATABASE DESIGN

The authors have created a test database containing three tables viz: DOCUMENT Table, KEYWORD Table, and DOC_KEY Table by inputting data relating to titles of conference papers on aerospace and associated descriptors in order to develop CODORS. Dialog OnDisc Aerospace Database forms the source for creating this test database. As is provided earlier each subject descriptor in the Dialog OnDisc Aerospace Database contains Broader Terms (BT) and Narrower Terms (NT) in addition to other descriptors. In the present investigation, the authors considered both these BT and NT on equal importance for representing the subject and for facilitating searches. The field specifications of three tables created in the test database are detailed below:

Document Table: Contains two fields such as ACC_NO and TITLE. The data type of ACC_NO is character of size 9 and TITLE is variable character of size 500. The primary key in DOCUMENT table is ACC_NO, which takes unique values assigned in Dialog OnDisc Aerospace Database and TITLE takes the document title as the value.

Keyword Table: KEYWORD table contains two fields such as KEY_NO and KEYWORD. The data type of KEY_NO is number of size 5 and KEYWORD is variable character of size 100. The primary key in KEYWORD table is KEY_NO, which takes unique incremental numeric values assigned by the researcher and KEYWORD takes unique descriptors as the value to avoid repetition of descriptors.

Doc_Key Table: DOC_KEY table contains three fields such as DOCKEY, ACC_NO and KEY_NO. The data type of DOCKEY is number of size 10, ACC_NO is variable character of size 9 and KEY_NO is number of size 5. The primary key in DOC_KEY table is DOCKEY, which takes unique incremental numeric values assigned by the researcher. ACC_NO takes values from DOCUMENT table and KEY_NO takes numeric values from KEYWORD table.

To ensure better efficiency and for faster retrieval of records, relationships were established between DOCUMENT and DOC_KEY tables on ACC_NO field. Similarly, relationships were established between KEYWORD and DOC_KEY tables on KEY_NO field.

9. DEVELOPMENT OF OPAC CLIENT SOFTWARE

The main purpose of developing OPAC client software is to compare the results retrieved using CODORS system and to draw some useful conclusions such as relevance ranking, percentage of retrieved results, etc. So, it was felt necessary to design and develop OPAC on the same test database of document titles. Authors have designed and developed OPAC client software on the developed test database in Oracle 9i using JSP programming language.

9.1 General OPAC Search Interface

In the present design and development of general OPAC software, the researcher has considered the query terms to be matched against the titles of conference papers. The search query is formed with the help of Boolean 'AND' operator in order to narrow the

Figure 1. General OPAC search interface.

search results. The OPAC search interface is shown in Fig. 1. The search interface allow the user to input search terms in text box, submit to retrieve results and clear entered text in text area.

9.2 General OPAC Search Results

When the user executes OPAC search by entering three terms such as 'ADVANCED', 'HYPERSONIC' and 'AIRCRAFT' separated by single space, OPAC search engine retrieves the results by matching titles in DOCUMENT table. OPAC search engine considers the matching of terms against document titles irrespective of the order of terms entered by the user. Figure 2 show two titles retrieved through general OPAC search.

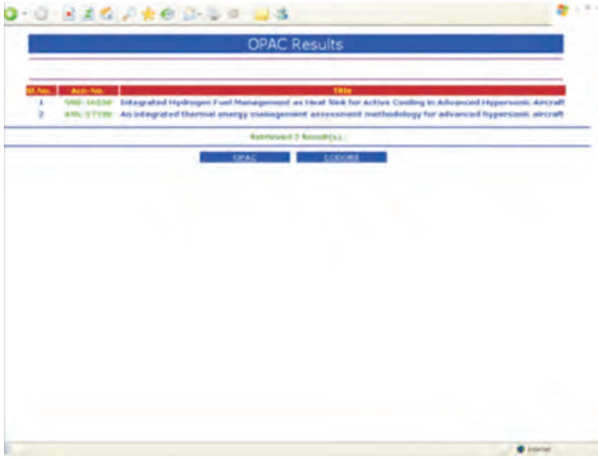

Figure 2. General OPAC search results.

10. DEVELOPMENT OF CODORS SOFTWARE

The authors have designed and developed CODORS software using JSP programming language on the developed test database in oracle 9i. In order to reduce the number of results, the investigator has formed the SQL (Structured Query Language) query with the help of Boolean 'AND' operator.

10.1 CODORS Flow Chart

The CODORS software has been developed based on the same database that is used for OPAC in order to study the comprehensiveness, relevance ranking, and relatedness of the results retrieved by both systems. Various steps involved in the development of CODORS are represented in the pictorial diagram in Fig. 3.

10.2 CODORS User Interface

In this design and development of CODORS software, the authors have considered the formation of query terms with the help of Boolean 'AND' operator in order to narrow the recommendations. The CODORS client interface, which accepts input from the user, is shown in Fig. 4.

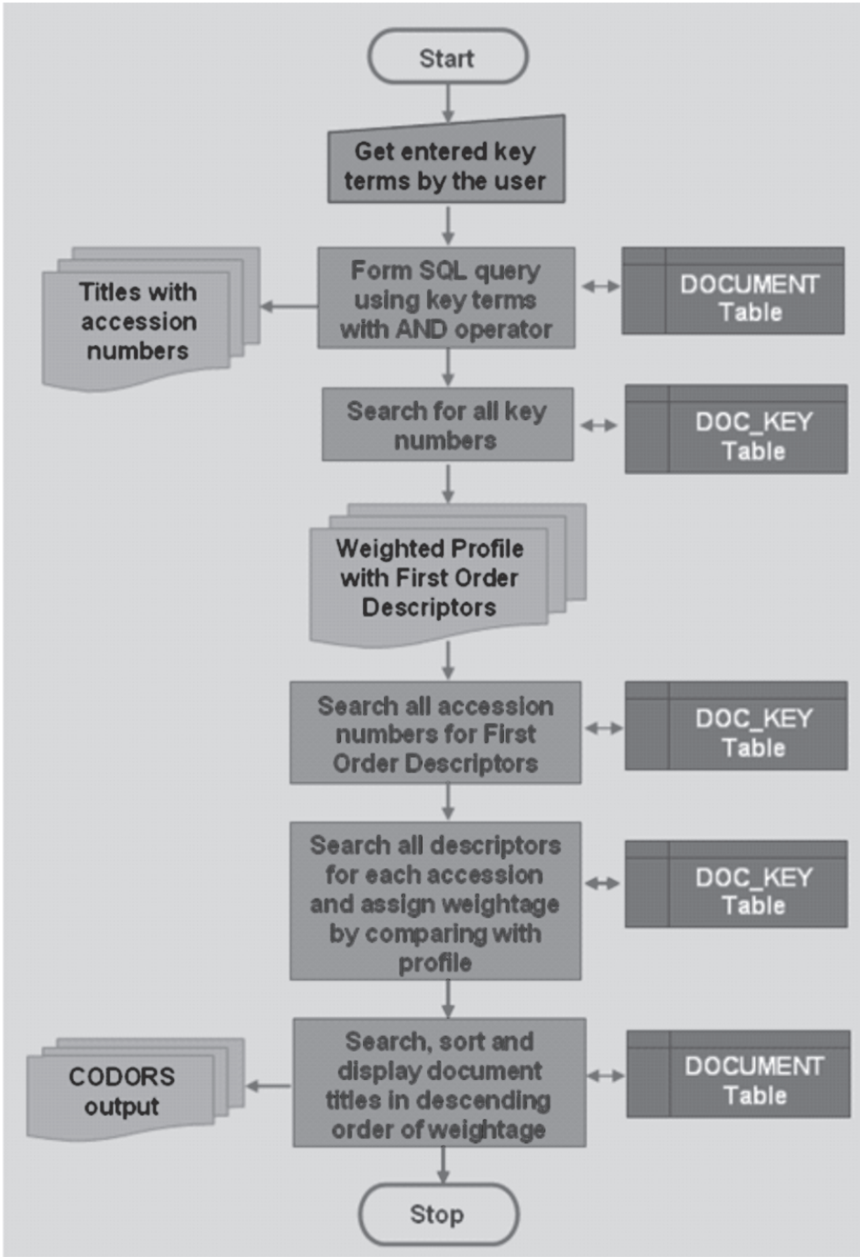

Figure 3. CODORS flowchart.

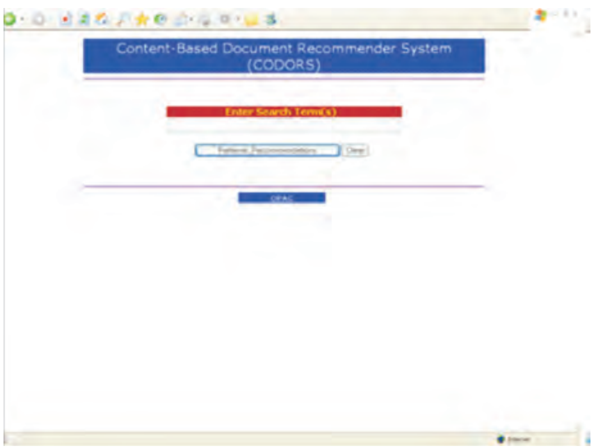

Figure 4. CODORS user interface.

10.3 Recommendation process in CODORS design

Step 1: Basic Input: In general, there exist different ways of obtaining input from the user for recommendation of items such as rating of sample database of items, choosing peer-group of users, provision of keywords or observing the navigation behavior of the user, etc. For example, Amalthaea[7,8], Bellcore Video Recommender[9], Eigentaste[10], Fab[11,12], Foxtrot Recommender System[13], ifWeb[14], LIBRA[15], PipeCF[16], PORSCHE[17], REFEREE[18], Ringo[19,20], SmartRadio[21,22] uses "numerical scale" for rating reference documents/items as input data collection method. DEMOIR[23] and Entrée[24,25] uses "positive"/"negative" method; InfoFinder[26], News Dude[27], and NewsWeeder[28] uses "interesting"/"not interesting" methods for input data collection. Movies2Go[29], SIFT Netnews[30], Tapestry[31] and WebSail[32,33] uses "like"/"dislike" methods to collect input data. Syskill & Webert[34] uses "hot"/"cold" method, GroupLens[35,36] uses "text comments" method for collection of input data and representation of profiles.

NCR News[37], AVATAR[38], Foxtrot Recommender System[13], NAUTILUS[39], Personal WebWatcher[40], PILGRIM[41], PORSCHE[17], REFEREE[18], SELECT[42], SurfLen[43], The Wasabi Personal Shopper[44], WebInEssence[45] and WebSIFT[46] uses "navigation behavior"/"browsing behavior" as input. Amazon.com[47], Beehive[48], GroupMark[49], LifestyleFinder[50] and VISCORS[51] uses historical data as input. Amalthaea[7,8], Anatagonomy[52], GroupLens[35,36], Krakatoa Chronicle[53], Recommendation Explorer[54], REFEREE[18] and SmartRadio[21,22] uses a form of "time spent"/ "idle time"/ "focus time"/ "saving"/ "scrolling"/ "enlarging"/ "printing"/ "downloading"/ "bookmarking", etc. for collection of input data.

In CODORS system, users are required to provide few useful and relevant keywords or terms. The user is expected to decide the relevance of these keywords for finding useful articles from the database. For retrieval of better results, the user can correlate with documents he has read previously and choose the terms to find such documents useful for his/her study. It is discouraged to input weak terms such as articles (like the, a, an) and words which do not carry the subject theme of documents.

The CODORS user interface accepts the input data from the user and forms SQL query with help of Boolean 'AND' operator to search the titles that match the term(s) entered by the user. Formation of SQL query is restricted to match the exact strings against the titles of papers. The titles retrieved at this stage are the basic input for building the user profile. For example, if the user enters three terms such as 'ADVANCED', 'HYPERSONIC', 'AIRCRAFT' as relevant terms separated by single space and submits for recommendations, CODORS recommendation engine retrieves the following two titles along with ACC_NO from DOCUMENT table of the database (Table 1).

Table 1. Results for 'ADVANCED', HYPERSONIC', 'AIRCRAFT' terms

Acc_No	Title
N98-16558	Integrated Hydrogen Fuel Management as Heat Sink for Active Cooling in Advanced Hypersonic Aircraft
A96-37738	An Integrated Thermal Energy Management Assessment Methodology for Advanced Hypersonic Aircraft

Step 2: Identification of Descriptors for Document Title: The above shown results are nothing but general OPAC search results that match terms entered by the user against title of the document. The CODORS engine moves to DOC_KEY table and retrieves KEY_NO and KEYWORD (descriptors) for documents having ACC_NOs already retrieved at the Step 1. KEY_NOs and KEYWORD (descriptors) for the documents having ACC_NO: 'N98-16558' and 'A96-37738' are shown in Table 2 and Table 3, respectively.

Table 2. Descriptors and its identification numbers for document having Acc_No= N98-16558

Key_No	Keyword
2954	COOLING
3799	DISSIPATION
6283	HEAT SINKS
6671	HYDROGEN FUELS
6734	HYPERSONIC AIRCRAFT
6749	HYPERSONIC VEHICLES
8576	MANAGEMENT SYSTEMS
12256	REGENERATIVE COOLING
14771	TEMPERATURE CONTROL

Table 3. Descriptors and its identification numbers for document having Acc_No= A96-37738

Key_No	Keyword
2954	COOLING SYSTEMS
5543	FUEL TANKS
6734	HYPERSONIC AIRCRAFT
7092	INLET TEMPERATURE
8576	MANAGEMENT SYSTEMS
14619	SYSTEMS ENGINEERING
14914	THERMAL ENERGY

Step 3: Generation of User Profile: Generation and maintenance of accurate user profile is the key for success of any recommender system. There are several ways of building and representing user profiles. For example, Bellcore Video Recommender[9], GroupLens[35,36], MovieReco[55], PocketLens[56], uses user-item ratings matrix, INFOrmer[57] uses weighted associative network, Entrée[24,25], ifWeb[14], InterestMap[58] and SiteIF[59] uses semantic networks, NAUTILUS[39] and Re: Agent[60] uses neural networks, InfoFinder[26] and WebSell[61,62] uses decision tree techniques, LaboUr[63], News Dude[27], Personal WebWatcher[40] and Syskill & Webert[34] use Bayesian classifiers, PSUN[64] uses weighted N-grams, Amalthaea[7,8], Anatagonomy[52], ARAS[65], Beehive[48], and CoFIND[66] uses vector space model for profile representation.

The CODORS recommendation engine utilizes descriptors retrieved at Step 2 with their multiple occurrences for generating weighted vector of user profile. When descriptors appear more than once, the relative ranking is decided by assigning weightages based on the number of occurrences of such descriptors. It is similar to the concept known as like-minded people share similar characteristics or documents that describe similar subject share more number of common descriptors. From the above retrieved descriptors it is clear that there are 16 descriptors of which ‘HYPERSONIC AIRCRAFT’, ‘MANAGEMENT SYSTEMS’ appeared twice and other descriptors appeared only once. The CODORS recommendation engine builds user profile using weighted vector-space model i.e. weighted vector of thesaurus descriptors as shown below.

$P = \left(w_1 d_1, w_2 d_2, w_3 d_3 .... w_i d_1\right)$, Where P is the user profile and $w_1, w_2, w_3,...w_i$ are weights associated with $d_1, d_2, d_3,...d_i$ descriptors.

Weights for the descriptors are calculated based on the number of occurrences of individual descriptor with reference to total number of descriptors retrieved at Step 2. These descriptors are called as ‘First Order Descriptors’. From the above example, total number of descriptors retrieved are 16 (including those occurred more than once) and each descriptor carries 6.25 as its weightage. Generated weighted vector of user profile is:

$$\textbf{User Profile} = \left(\begin{array}{l} 6.25 * COOLING, \\ 6.25 * COOLING \ldots SYSTEMS, \\ 6.25 * DISSIPATION, \\ 6.25 * FUEL \ldots TANKS, \\ 6.25 * HEAT \ldots SINKS, \\ 6.25 * HYDROGEN \ldots FUELS), \\ 12.5 * HYPERSONIC \ldots AIRCRAFT, \\ 6.25 * HYPERSONIC \ldots VEHICLES, \\ 6.25 * INLET \ldots TEMPERATURE, \\ 12.5 * MANAGEMENT \ldots SYSTEMS, \\ 6.25 * REGENERATIVE \ldots COOLING, \\ 6.26 * SYSTEMS \ldots ENGINEERING, \\ 6.25 * TEMPERATURE \ldots CONTROL, \\ 6.25 * THERMAL \ldots ENERGY \end{array} \right)$$

Step 4: Extraction of documents having first order descriptors: At this stage, using DOC_KEY and DOCUMENT tables, the CODORS recommendation engine extracts all document titles having atleast any one of the “First Order Descriptors”. As a result retrieved titles will be more in number compared to general OPAC results, which only matches titles with the keyword input by the user.

Step 5: Calculating relevance ranking for results retrieved at Step 4: The CODORS recommendation engine further performs search against DOC_KEY table to find out number of descriptors assigned to each and every document retrieved at Step 4. The descriptors of each document are compared with the ‘First Order Descriptors’ or “Weighted Vector of Descriptors” mentioned at Step 3 and its final weightage is calculated by summing up of weights of matched descriptors. In the above given example, CODORS recommendation engine retrieved 6615 documents from the database which are having atleast one of the “First Order Descriptors”. Calculation of percentage of relevance by comparing with weighted vector of user profile is described for first five documents out of 6615 and is shown in below:

1. Acc_No	**N98-16558**
TITLE	Integrated Hydrogen Fuel Management as Heat Sink for Active Cooling in Advanced Hypersonic Aircraft
DESCRIPTORS & PERCENTAGE OF THEIR RELEVANCE WEIGHTAGE	6.25 * COOLING 6.25 * DISSIPATION 6.25 * HEAT SINKS 6.25 * HYDROGEN FUELS 12.5 * HYPERSONIC AIRCRAFT 6.25 * HYPERSONIC VEHICLES 12.5 * MANAGEMENT SYSTEMS 6.25 * REGENERATIVE COOLING 6.25 * TEMPERATURE CONTROL
TOTAL PERCENTAGE RELEVANCE WEIGHTAGE	68.75 %

2. Acc_No	**A96-37738**
TITLE	An integrated thermal energy management assessment methodology for advanced hypersonic aircraft
DESCRIPTORS & PERCENTAGE OF THEIR RELEVANCE WEIGHTAGE	6.25 * COOLING SYSTEMS 6.25 * FUEL TANKS 12.5 * HYPERSONIC AIRCRAFT 6.25 * INLET TEMPERATURE 12.5 * MANAGEMENT SYSTEMS 6.25 * SYSTEMS ENGINEERING 6.25 * THERMAL ENERGY
TOTAL PERCENTAGE RELEVANCE WEIGHTAGE	56.25 %

3. Acc_No	N97-17338
TITLE	An Integrated Thermal Management Analysis Tool
DESCRIPTORS & PERCENTAGE OF THEIR RELEVANCE WEIGHTAGE	0.00 * AERODYNAMIC CONFIGURATIONS 0.00 * AERODYNAMIC HEATING 0.00 * AIRCRAFT DESIGN 0.00 * AIRCRAFT ENGINES 0.00 * AIRFRAMES 0.00 * APPLICATIONS PROGRAMS (COMPUTERS) 0.00 * CIRCUITS 6.25 * COOLING SYSTEMS 0.00 * ENGINE PARTS 6.25 * FUEL TANKS 0.00 * GRAPHICAL USER INTERFACE 0.00 * HEAT EXCHANGERS 0.00 * MANAGEMENT ANALYSIS 12.5 * MANAGEMENT SYSTEMS 0.00 * MASS FLOW RATE 0.00 * SATELLITE TEMPERATURE 0.00 * SPACECRAFT DESIGN 0.00 * STRUCTURAL MEMBERS 0.00 * SYSTEMS INTEGRATION 6.25 * TEMPERATURE CONTROL 0.00 * TEMPERATURE DISTRIBUTION 0.00 * TEMPERATURE EFFECTS
TOTAL PERCENTAGE RELEVANCE WEIGHTAGE	31.25%

4. Acc_No	A98-22754
TITLE	A structured approach to software interface development for the V-22 Osprey
DESCRIPTORS & PERCENTAGE OF THEIR RELEVANCE WEIGHTAGE	0.00 * AIRCRAFT CONTROL 0.00 * COCKPITS 6.25 * FUEL TANKS 12.5 * MANAGEMENT SYSTEMS 0.00 * SOFTWARE TOOLS 6.25 * SYSTEMS ENGINEERING 0.00 * SYSTEMS INTEGRATION 0.00 * V-22 AIRCRAFT
TOTAL PERCENTAGE RELEVANCE WEIGHTAGE	25.0%

5. Acc_No	N94-23636
TITLE	An engineering code to analyse hypersonic thermal management systems
DESCRIPTORS & PERCENTAGE OF THEIR RELEVANCE WEIGHTAGE	0.00 * AIRCRAFT DESIGN 0.00 * COMPUTER PROGRAMS 0.00 * CONVECTIVE FLOW 0.00 * ENERGY DISSIPATION 0.00 * HEAT FLUX 6.25 * HYPERSONIC VEHICLES 0.00 * HYPERSONICS 12.5 * MANAGEMENT SYSTEMS 0.00 * OPTIMIZATION 0.00 * SPACECRAFT DEFENSE 6.25 * TEMPERATURE CONTROL
TOTAL PERCENTAGE RELEVANCE WEIGHTAGE	25.0%

Step 6: Estimation of Boosting Factor: Retrieval of 100 per cent relevance document(s) at Step 4 is an ideal situation and it is possible only when one document is retrieved at Step 1 or same number of descriptors is retrieved at step 2 for more than one document. In other situations, it is not possible to retrieve 100% relevance document(s) because of sharing of different descriptors. In such cases, authors raised the numeric value of its relevance weightage to 100 percent by adding additional weightage points. This factor of adding additional weightage points is terms as “Boosting Factor”. Thus:

Boosting Factor = 100 - Highest Relevance Weighted Document Points

In the example given above the following document has highest relevance weightage points (Acc_No=N98-16558):

ACC_NO	Title	Relevance points
N98-16558	Integrated Hydrogen Fuel Management as Heat Sink for Active Cooling in Advanced Hypersonic Aircraft	68.75

Boosting Factor of the said document (to take it to 100 per cent relevance weightage) is calculated as: Boosting Factor = 100 - 68.75 = 31.25.

This Boosting Factor is added to the relevance weightage of every document so that the most relevant document carry 100 per cent relevance weightage while displaying the results and other documents are also boosted to the same level of relevance. Table 4 shows actual relevance weightage, boosting factor and final relevance weightage for first few documents.

Table 4. Actual relevance weightage, boosting factor and final relevance weightage for few documents

Acc_No	Actual relevance weightage	Boosting factor	Final relevance weightage	Relevance rank
N98-16558	68.75	31.25	100.00	1
A96-37738	56.25	31.25	87.50	2
N97-17338	31.25	31.25	62.50	3
A98-22754	25.00	31.25	56.25	4
N94-23636	25.00	31.25	56.25	4
A98-27896	18.75	31.25	50.00	5
A98-27056	18.75	31.25	50.00	5
N97-28096	18.75	31.25	50.00	5
A97-39037	18.75	31.25	50.00	5

Step 7: Sorting of recommendations: For further study and analysis of CODORS recommendations, the results obtained at Step 6 are sorted on decreasing order of relevance ranking. Table 5 shows sorting of CODORS recommendations for display for the user seeking for recommendations of technical documents.

Step 8: Display of recommendations: In the 8th and final step, CODORS retrieves all documents in a descending order from the first ranked document with 100 per cent relevance weightage (after adding boosting factor in case the highest relevance weightage of retrieved documents is (are) less than 100 per cent) to the last document having least percentage of relevance weightage.

The user can have cut-off point depending upon the number of documents he/she wants to use because these documents are arranged in descending order of relevance (weightage). In the above example the total hit documents were 6615. However, Fig. 5 lists five documents ranked in descending order with cut-off point at greater than 50 percentage relevance weightage.

Table 5. Decreasing order of recommendations by CODORS

Acc_No	Actual relevance rank	Boosting factor	Final relevance rank
N98-16558	68.75	31.25	100.00
A96-37738	56.25	31.25	87.50
N97-17338	31.25	31.25	62.50
A98-22754	25.00	31.25	56.25
N94-23636	25.00	31.25	56.25
A98-27896	18.75	31.25	50.00
A98-27056	18.75	31.25	50.00
N97-28096	18.75	31.25	50.00
A97-39037	18.75	31.25	50.00
A97-44081	18.75	31.25	50.00
N93-22846	18.75	31.25	50.00
N95-16322	18.75	31.25	50.00
N93-22608	18.75	31.25	50.00
A00-21313	18.75	31.25	50.00
A95-34431	18.75	31.25	50.00
N92-26987	18.75	31.25	50.00
N95-14555	18.75	31.25	50.00
A97-14633	18.75	31.25	50.00
N95-27354	18.75	31.25	50.00
N95-17596	18.75	31.25	50.00
N94-23190	18.75	31.25	50.00
N92-27020	18.75	31.25	50.00
N94-29285	18.75	31.25	50.00
A98-11230	18.75	31.25	50.00
N95-13892	18.75	31.25	50.00
N95-14567	18.75	31.25	50.00
N93-27844	18.75	31.25	50.00
N95-17933	18.75	31.25	50.00
N92-18620	18.75	31.25	50.00
N95-17585	18.75	31.25	50.00
A97-44456	18.75	31.25	50.00
N93-26937	18.75	31.25	50.00
A98-34472	18.75	31.25	50.00
N96-22509	18.75	31.25	50.00
A00-26112	18.75	31.25	50.00
N92-27898	18.75	31.25	50.00
N92-27893	18.75	31.25	50.00
N92-26985	18.75	31.25	50.00
N94-24413	18.75	31.25	50.00
N97-11488	18.75	31.25	50.00
A97-28029	18.75	31.25	50.00
N92-17101	18.75	31.25	50.00
N92-16481	18.75	31.25	50.00

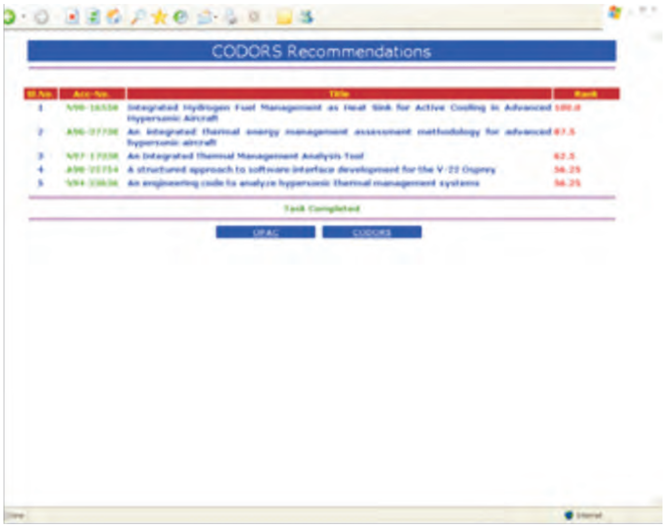

Figure 5. CODORS recommendations.

11. CONCLUSION

Every recommender system needs some sort of input collected explicitly or implicitly to generate recommendations. Using this input, the recommender system builds user profile which represents any of the IR/profile representation models viz., vector space model, semantic networks, weighted n-grams, associative networks etc. The recommender system matches user profile with the descriptions of items or users with similar taste to recommend items of interest to the user. There are three matching techniques generally used by the recommender systems such as: content-based filtering, collaborative filtering, demographic filtering or hybrid technique. Finally, these systems provide Top-N recommendations followed by fine tuning the user profile with the help of relevance feedback mechanism to retrieve more relevant items. Hybrid recommender systems are more important and they are going to play vital role in the next-generation of recommender systems, because they minimize the disadvantages and maximize the utility of content-based filtering systems and collaborative filtering systems.

Presently designed COntent-based Document Recommender System (CODORS) converts the terms expressed by the user in natural language automatically into subject descriptors, carry on search, rank documents and retrieve. The document ranking is automatically done by the system on the basis of weightages calculated based on the occurrences of number of subject descriptors, which are assigned to each title of the document in the collection.

The results are sorted on relative relevance ranking and are presented to the user for maximum utilization of technical resources that are otherwise hidden in the database collection. CODORS was put to test by conducting repeated searches using variety of descriptors/key terms relating to general and specialized subject areas in the field of aerospace engineering. The results exhibited that the CODORS search provided many more relevant documents and increased the recall value as compared to general OPAC search.

REFERENCES

1. Furnas, G.W.; Landauer, T.K.; Gomez, L.M. & Dumais, S.T. Statistical semantics: Analysis of the potential performance of keyword information systems. *Bell Sys. Tech. J.*, 1983, **62**(6), 1753-806.

2. Hildreth, C.R. Pursuing the ideal: Generations of online catalogs, in Online catalogs, online reference: Converging trends, edited by B. Aveney and B. Butler. *In* Proceedings of a Library and Information Technology Association Pre-conference Institute. American Library Association, Chicago, 1983. pp. 31-56.

3. Hildreth, C.R. Beyond Boolean: Designing the next generation of online catalogs. *Library Trends*, 1987, **35**(4), 647-67.

4. Hildreth, C.R. Intelligent interfaces and retrieval methods for subject searching in bibliographic retrieval systems. Washington, DC, Library of Congress, 1989.

5. Hildreth, C.R. OPAC research: Laying the groundwork for future OPAC design, edited by C.R. Hildreth. *In* The online catalogue: Development and directions. Library Association, London, 1989. pp.1-24.

6. Hildreth, C.R. Review—enhancing access to information: Designing catalogs for the 21st century. *Inf. Process. Manag.*, 1997, **33**(4), 573-75.

7. Moukas, A. & Maes, P. Amalthaea: An evolving multi-agent information filtering and discovery system for the WWW. *Autonomous Agents and Multi-Agent Syst.,* 1998, **1**(1), 59-88.

8. Moukas, A. Amalthaea: Information discovery and filtering using a multiagent evolving ecosystem. *App. Arti. Intelli.: An Int. J.*, 1997, **11**(5), 437-57.

9. Hill, W.; Stead, L.; Rosenstein, M. & Furnas, G. Recommending and evaluating choices in a virtual community of use, edited by I.R. Katz, R. Mack, L. Marks, M.B. Rosson & J. Nielsen. *In* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado, United States. ACM Press, New York, 1995. pp.194-201.

10. Goldberg, K.; Roeder, T.; Gupta, D. & Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval,* 2001, **4**(2), 133-51.

11. Balabanovic, M. & Shoham, Y. Fab: Content-based, collaborative recommendation. *Communications of ACM*, 1997, **40**(3), 66-72.

12. Balabanovic, M. An adaptive webpage recommendation service. *In* Proceedings of 1st International Conference on Autonomous Agents, Marina del Rey, California, United States. ACM Press, New York, 1997. pp. 378-85.

13. Middleton, S.E.; De Roure, D.C. & Shadbolt, N.R. Foxtrot recommender system: User profiling, ontologies and the World Wide Web. *In* Proceedings of the 11th International World Wide Web Conference, Honolulu, Hawaii, 2002.

14. Asnicar, F. & Tasso, C. ifWeb: A prototype of user model-based intelligent agent for documentation filtering and navigation in the World Wide Web. *In* Proceedings of the 6th International Conference on User Modelling, Chia Laguna, Sardinia, 1997. pp. 3-12.

15. Mooney, R.J. & Roy, L. Content-based book recommending using learning for text categorization. *In* Proceedings of the 5th ACM Conference on Digital Libraries, San Antonio, Texas. ACM Press, New York, 2000. pp. 95-204.

16. Xie, B.; Han, P. & Shen, R. PipeCF: A scalable DHT-based collaborative filtering recommendation system, edited by S.I. Feldman, M. Uretsky, M. Najork, & C.E. Wills. *In* Proceedings of the 13th International Conference on World Wide Web—Alternate Track Papers & Posters, New York, USA, 2004. pp. 224-25.

17. Kawashima, H.; Matsushita, T. & Satake, S. PORSCHE: A physical objects recommender system for cell phone users. *In* Proceedings of the 2nd International Workshop on Personalised Context Modeling and Management for UbiComp Applications, Orange County, California, 2006.

18. Cosley, D.; Lawrence, S. & Pennock, D.M. Referee. An open framework for practical testing of recommender systems using ResearchIndex. *In* Proceedings of 28th International Conference on Very Large Data Bases, Hong Kong, China, 2002. pp. 35-46.

19. Shardanand, U. Social information filtering for music recommendation. Program in Media Arts and Sciences, Massachusetts Institute of Technology, 1994. (M.S. Thesis).

20. Shardanand, U. & Maes, P. Social information filtering: Algorithms for automating "word of mouth", edited by I.R. Katz, R. Mack, L. Marks, M.B. Rosson, & J. Nielsen. *In* Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Denver, Colorado, United States. ACM Press/Addison-Wesley, 1995. pp. 210-17.

21. Hayes, C., & Cunningham, P. Smart radio-Building music radio on the fly. Proceedings of Expert Systems 2000, edited by A. Macintosh; M. Moulton & F. Coenen. *In* 20th SGES Conference on KBS and Applied AI. Springer Verlag, London, 2000. pp.129-40.

22. Hayes, C. & Cunningham, P. Smart Radio—Community-based music radio. *Knowledge Based Syst.*, 2001, **14**(3-4), 197-201.

23. Yimam-Seid, D. & Kobsa, A. Expert finding systems for organizations: Problem and domain analysis and the DEMOIR approach. *J. Organis. Comput. Electr. Commerce,* 2003, **13**(1), 1-24.

24. Burke, R.; Hammond, K. & Cooper, E. Knowledge-based navigation of complex information spaces. *In* Proceedings of the 13th National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference, Portland, AAAI Press/The MIT Press, Oregon, 1996. pp. 462-68.

25. Burke, R.; Hammond, K. & Young, B. The FindMe approach to assisted browsing. *IEEE Expert*, 1997, **12**(4), 32-40.

26. Krulwich, B. & Burkey, C. The InfoFinder agent: Learning user interests through heuristic phrase extraction. *IEEE Expert: Intelli. Syst. Appl.,* 1997, **12**(5), 22-27.

27. Billsus, D. & Pazzani, M.J. A hybrid user model for news story classification, edited by J. Kay. *In* Proceedings of the 7th International Conference on User Modeling, Banff, Canada. Springer-Verlag, New York, 1999. pp. 99-108.

28. Lang, K. NewsWeeder: Learning to filter netnews, edited by Armand Prieditis & Stuart J.Russell. *In* Proceedings of the 12th International Conference on Machine Learning, Tahoe City, California, USA, 1995. pp. 331-39.

29. Mukherjee, R.; Jonsdottir, G.; Sen, S. & Sarathi, P. MOVIES2GO: An online voting based movie recommender system. *In* Proceedings of the 5th international Conference on Autonomous Agents, Montreal, Quebec, Canada. ACM Press, New York, 2001. pp. 14-15.

30. Yan, T.W. & Garcia-Molina, H. SIFT—A tool for wide-

area information dissemination. *In* Proceedings of USENIX 1995 Technical Conference on UNIX and Advanced Computing Systems, New Orleans, Louisiana. Berkeley, USENIX Association, CA, 1995. pp.177-86.

31. Goldberg, D.; Nichols, D.; Oki, B.M. & Terry, D. Using collaborative filtering to weave an information tapestry. *Communications of ACM*, 1992, **35**(12), 61-70.

32. Chen, Z.; Meng, X.; Fowler, R.H. & Zhu, B. WebSail: From online learning to Web search, edited by Q. Li, Z.M. Ozsoyoglu, R. Wagner, Y. Kambayashi, & Y. Zhang. *In* Proceedings of the 1st International Conference on Web Information Systems Engineering, Hong Kong, China. Washington, DC, IEEE Computer Society, 2000. pp. 206-13.

33. Chen, Z.; Meng, X.; Zhu, B. & Fowler, R.H. WebSail: From online learning to Web search. *Know. Inf. Syst.,* 2002, **4**(2), 219-27.

34. Pazzani, J.M. & Billsus, D. Syskill and webert: Identifying interesting websites. *In* Proceedings of the 13th National Conference on Artificial Intelligence and 8th Innovative Applications of Artificial Intelligence Conference, Portland, AAAI Press/The MIT Press, Oregon, 1996. pp. 54-61.

35. Konstan, J.A.; Miller, B.N.; Maltz, D.; Herlocker, J.L.; Gordon, L.R. & Riedl, J. GroupLens: Applying collaborative filtering to Usenet news. *Communications of ACM*, 1997, **40**(3), 77-87.

36. Resnick, P.; Iacovou, N.; Suchak, M.; Bergstrom, P. & Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. *In* Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, Chapel Hill, North Carolina, United States. ACM Press, New York, 1994. pp. 175-86.

37. Mobasher, B.; Cooley, R. & Srivastava, J. Automatic personalisation based on Web usage mining. *Communications of the ACM*, 2000. **43**(8), 142-51.

38. Blanco-Fernandez, Y.; Pazos-Arias, J.J.; Lopez-Nores, M.; Gil-Solla, A. & Ramos-Cabrer, M. AVATAR: An improved solution for personalized TV based on semantic interface. *IEEE Trans. Cons. Electro.,* 2006, **52**(1), 223-31.

39. Gori, M.; Maggini. M. & Martinelli, E. NAUTILUS: Navigate Autonomously and Target Interesting Links for Users. Technical report no. RT-DII-20/99. Dipartimento di Ingegneria dell'Informazione, Università di Siena, Italy, 1999.

40. Mladenic, D. Personal WebWatcher: Design and implementation. Technical Report No. IJS-DP-7472, School of Computer Science, Carnegie-Mellon University, Pittsburgh, USA, 1996.

41. Brunato, M. & Battiti, R. PILGRIM: A location broker and mobility-aware recommendation system. *In* Proceedings of the 1st IEEE International Conference on Pervasive Computing & Communications, Fort Worth(TX), USA. IEEE Computer Society, Washington, DC, 2003. pp. 265-72.

42. Alton-Scheidl, R.; Ekhall, J.; van Geloven, O.; Kovacs, L.; Micsik, A.; Lueg, C., *et al*. SELECT: Social and collaborative filtering of web documents and news, edited by A. Kobsa & C. Stephanidis. *In* Proceedings of the 5th ERCIM Workshop on User Interfaces for All, Dagstuhl, Germany, 1999. pp. 23-37.

43. Fu, X.; Budzik, J. & Hammond, K.J. Mining navigation history for recommendation. *In* Proceedings of the 5th International Conference on Intelligent User Interfaces, New Orleans, Louisiana, United States. ACM Press, New York, 2000. pp.106-12.

44. Burke, R. The Wasabi Personal Shopper: A case-based recommender system. *In* Proceedings of the Sixteenth National Conference on Artificial Intelligence and the 11th Innovative Applications of Artificial Intelligence Conference, Orlando, Florida, USA. Menlo Park, AAAI, CA, 1999. pp. 844-49.

45. Radev, D.R.; Fan, W. & Zhang, Z. WebInEssence: A personalised web-based multi-document summarisation and recommendation system. *In* Proceedings of NAACL Workshop on Automatic Summarisation, Pittsburgh, 2001. pp. 79-88.

46. Cooley, R.; Mobasher, B. & Srivastava, J. Web mining: Information and pattern discovery on the World Wide Web. *In* Proceedings of 9th International Conference on Tools with Artificial Intelligence, Newport Beach, CA, USA. IEEE Computer Society, Washington, DC, 1997. pp. 558-67.

47. Linden, G.; Smith, B. & York, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 2003, **7**(1), 76-80.

48. Huberman, B.A. & Kaminsky, M. Beehive: A system for cooperative filtering and sharing of information. (Technical Report). Dynamics of Computation Group, Xerox Palo Alto Research Centre, 1996.

49. Pemberton, D.; Rodden, T. & Procter, R. GroupMark: A WWW recommender system combining collaborative and information filtering, edited by P.L.

Emiliani & C. Stephanidis. *In* Proceedings of 6th ERCIM Workshop on 'User Interfaces for All', Florence, Italy, 2000.

50. Krulwich, B. Lifestyle finder: Intelligent user profiling using large-scale demographic data. *Arti. Intelli. Mag.*, 1997, **18**(2), 37-46.

51. Kim, C.Y.; Lee, J.K.; Cho, Y.H. & Kim, D.H. VISCORS: A visual-content recommender for the mobile Web. *IEEE Intelli. Syst.*, 2004. **19**(6), 32-39.

52. Sakagami, H. & Kamba, T. Learning personal preferences on online newspaper articles from user behaviours, edited by P. H. Enslow, M. Genesereth, & A. Patterson. *In* Selected Papers from 6th International Conference on World Wide Web, Santa Clara, California. Elsevier Science Publishers, Essex, 1997. pp. 1447-455.

53. Kamba, T.; Bharat, K. & Albers, M.C. The Krakatoa chronicle: An interactive personalised newspaper on the Web. *In* Proceedings of 4th International World Wide Web Conference, Boston, USA, 1995. pp.159-70.

54. Efron, M. & Geisler, G. Is it all about connections? Factors affecting the performance of a link-based recommender systems. *In* Proceedings of ACM SIGIR Workshop on Recommender Systems, New Orleans, USA, 2001.

55. Medhi, D.G. & Dakua, J. MovieReco: A recommendation system. *Proceed. World Acad. Sci. Eng. Technol.*, 2005. **4**, 70-73.

56. Miller, B.N.; Konstan, J.A. & Riedl, J. PocketLens: Toward a personal recommender system. *ACM Trans. Inf. Syst.*, 2004, **22**(3), 437-76.

57. Sorensen, H.; O'Riordan, A. & O'Riordan, C. Profiling with the INFOrmer text filtering agent. *J. Univ. Comp. Sci.*, 1997, **3**(8), 988-1006.

58. Liu, H. & Maes, P. InterestMap: Harvesting social network profiles for recommendations. *In* Proceedings of IUI Beyond Personalisation 2005: A Workshop on the Next Stage of Recommender Systems Research, San Diego, USA, 2005. pp. 54-59.

59. Stefani, A. & Strapparava, C. Personalising access to websites: The SiteIF project. *In* Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, Pittsburgh, USA, 1998.

60. Boone, G. Concept features in Re: Agent, an intelligent email agent, edited by K.P. Sycara & M. Wooldridge. *In* Proceedings of the Second International Conference on Autonomous Agents, Minneapolis, Minnesota, USA. ACM, New York, 1998. 141-48.

61. Cunningham, P.; Bergmann, R.; Schmitt, S.; Traphoner, R.; Breen, S. & Smyth, B. WEBSELL: Intelligent sales assistants for the World Wide Web. *Kunstliche Intelligenz (KI),* 2001, **1**, 28-32.

62. Cunningham, P.; Bergmann, R.; Schmitt, S.; Traphoner, R.; Breen, S. & Smyth, B. Intelligent support for online sales: The WEBSELL experience, edited Weber, & C.G von Wangenheim. *In* Proceedings of the Workshop Programme at the Fourth International Conference on Case-Based Reasoning, Vancouver, British Columbia, Canada, 2001. pp.104-09.

63. Schwab, I.; Pohl, W. & Koychev, I. Learning to recommend from positive evidence. *In* Proceedings of the 5th International Conference on Intelligent User Interfaces, New Orleans, Louisiana, United States. ACM Press, New York, 2000. pp. 241-47.

64. Sorensen, H. & McElligott, M. PSUN: A profiling system for Usenet news. *In* CIKM'95 Workshop on Intelligent Information Agents, Baltimore, 1995.

65. Kilfoil, M.; Xing, W. & Ghorbani, A. ARAS: Adaptive recommender for academic scheduling. *In* Proceedings of the 3rd Annual Communication Networks and Services Research Conference. IEEE Computer Society, Washington, 2005. pp. 307-12.

66. Dron, J. Mitchell, R.; Siviter, P. & Boyne, C. CoFIND-An experiment in N-dimensional collaborative filtering. *J. Network Comp. Appli.*, 2003, **23**(2), 131-42.

About the Author

Dr K. Nageswara Rao received Masters Degree in Physics from Sri Venkateswara University, Tirupati; PGDCA from Jawaharlal Nehru Technological University, Hyderabad; Masters Degree in Library and Information Science in 1992 from Annamalai University; and PhD from University of Mysore. From 1993 to 1995, he worked at the National Informatics Centre, Hyderabad. During 1995 to 1999, he worked as Scientist at the Naval Physical & Oceanographic Laboratory, Kochi. Presently, he is Head Knowledge Centre, Defence Research & Development Laboratory, Hyderabad. His research interest encompasses Internet and information technologies for library applications, Web mining, design and development of digital libraries and recommender systems, personalisation and alerting services. He has published/presented a number of technical papers in national/international journals/conferences.