ADDIS ABABA UNIVERSITY
SCHOOL OF GRAGUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

APPLICATION OF COLLABORATIVE FILTERING AGENT FOR DOCUMENT RECOMMENDATION IN SDI SYSTEM

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Masters of Science in Information Science

By:
ZEHARA ZINAB ABABOR
July, 2003

ADDIS ABABA UNIVERSITY
LIBRARIES
P O BOX 1176
ADDIS ABABA ETHIOPIA

ADDIS ABABA UNIVERSITY
SCHOOL OF GRAGUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

APPLICATION OF COLLABORATIVE FILTERING AGENT FOR DOCUMENT RECOMMENDATION IN SDI SYSTEM

By:
ZEHARA ZINAB ABABOR
July, 2003

Name and signature of members of the examining board

Name	**Signature**
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Dedicated to

Zinab Ababor (My Dear Father)

and

Asrat Feleke (My Beloved Husband)

Acknowledgment

My greatest thanks goes to my advisors A to Tesfaye Biru, A to Workshet Lamenew and W/t Ethiopia Taddese. Without their advice and follow-up this research would not have been possible. They used to inspire and encourage me all the times especially at the most difficult and seemingly hopeless situations.

I am also grateful to all staff members of ILRI Information service particularly W/r Maria Bruneri and Ato Getachew Bultefa. I would also like to thank Ato Salah Yusuf for his cooperation.

My special thanks goes to my husband, Asrat Feleke, for his care, support and inspiration. Without his full support neither the research nor the two years study would have been possible. Last but not least, big thanks to all classmates for their support and cooperation throughout the study period at the school.

Abstract

Selective Dissemination of Information (SDI) is a personalized information filtering method used for delivering current information to users. The objective of SDI service is to reduce users' effort in their pursuit of relevant information. However, the performance of SDI service based on simple matching of user interest profiles with new documents, has been inefficient.

To improve the efficiency of such services agent based approaches are proposed. Collaborative filtering system is one type of agent technology that uses a database about user preferences to predict additional documents of items that the new user might like. It combines the opinions of humans to make personalized, accurate predictions. This study is therefore, an attempt to assess the applicability of collaborative filtering approach to SDI service.

This report presents the basic ideas of collaborative filtering systems and describes the results obtained from a preliminary experiment conducted based on the data obtained from International Livestock Research Institute's (ILRI) SDI service. Finally, based on the experiment findings, recommendations for further research are forwarded.

Table of Contents

List of Figures

List of Tables

Chapter One

Introduction

1.1. Background

1.1.1. Selective Dissemination of Information and Information Filtering

This era is characterized by the explosive growth in the amount of information generated and in the demand for information. Edmunds and Morries (2000) described the 'information age' as the age in which people are bombarded with information whether or not they seek it.

Among the things that are believed to contribute to this exponential growth of information are: the industrial and commercial applications (and the increased research activities as a result) and the technological development (Pao, 1989; Rowely and Tunner, 1987; Edmunds and Morries, 2000).

The information overload (or information explosion as some prefer to call it) is a world wide issue since "keeping up with the important findings in one's field as well as managing that information has become an increasingly important and challenging aspect of the career of a scientist" (Pao, 1989). This problem of keeping up with the important findings is the result of the increasing volume of scientific literature, which has created significant problems in information retrieval (Neill, 1992). Neill argued that this problem of information retrieval is fueled by two reasons: one is the problem of relevance i.e., a person may face a number of irrelevant information while trying to retrieve information of interest. The other, quite related to relevance, is that the person may encounter outdated information. In addition to this, as a result of information overload, there is also a problem of nonuse of available information.

The nonuse of information causes redundancy and duplication of efforts. Two or more researchers may conduct same research using the same methodology at the same time. Similarly, if the largely available information sources are not discovered and used by researchers/scientist, Wilson (1995) argued, their work will be inefficient and irrational. According to Wilson, "overload may come to be seen as more than a mere inconvenience and rather as a real threat to the rationality of the research process. If overload is as wide spread as the constant testimony of scientist and scholars claim it is, then research and development would deem to be extensively inefficient and irrational."

Furthermore, the physiological and intellectual capacity of human beings is so limited that it is difficult to read the vast amount of information that deal even with a single topic. As Neill (1992) noted, "there is more information on most subjects that can be read by one person in any reasonable amount of time."

In order to tackle these problems encountered as a result of information overload, new ways of handing information are required. Workers in the area must find ways to respond to the specific and urgent information needs. As Bose (1986) argued, "the vast output of scientific and technological information generated by multiple media needs to be effectively controlled for its dissemination."

So as to effectively control the dissemination of information and to help end-users (i.e. information users) discover the right information, new mechanisms and services should be provided. S elective D issemination o f Information (SDI) service is one frequently cited way of addressing such issues (Ferreira and Silva, 2001; Yan and Garcia-Molina, 1997).

SDI (also known as alerts or alerting service) as defined by Kemp (1979), "involves the provision of a service where by the individual client is sent notification of items in the literature which match a statement of his requirements called a 'profile'. A computer is used to compare records of documents with the profiles of clients and to print the notification." This shows that common SDI services are computer-based services. The notification can also be made through electronic mail.

Packer and Soergel (1979) also pointed out that, SDI is "a service that tackles the information problem by keeping him (the scientist) continuously informed of new documents published in his areas of the latest developments." The objective of an SDI service, as pointed out by Rao (1990), is to keep users of the system informed of the developments in their respective fields of interest. This must be done without deluging the users with non-relevant and unwanted documents. The goal is to provide as few non-relevant documents and as much relevant documents as possible.

SDI, as stated by Ferreira and Silva (2001), is an asynchronous service composed of two complementary workflows. It first identifies and classifies information relevant to users, from heterogeneous and continuously updated sources of information. And then, notifies users about those significant results and record their feedbacks for refinement of their needs. Relevant information or documents are identified and classified based on the long lasting interest of users. This long lasting interest is called interest profile.

Thus, the problem of SDI is related with the issue of Information Filtering (IF) and Information Retrieval (IR) (Ferreira and Silva, 2001). These systems (IR and IF) are basically interested in selection of a subset of documents from the document set, which are relevant to a user interest (Griffith and O'Riordan, 2000).

The difference between IR and IF is usually taken to be a difference in the nature of the data and the nature of the user request. With IR, the data set is relatively static and the user request is a one-off query. With IF, the data set is dynamic and the user request is a profile representing a long term interest (Sarwar et al, 1998).

There are two commonly cited classes of information filtering: *cognitive*, based on the content of articles (content based filtering) and *social* (collaborative filtering) (Griffith and O'Riordan, 2000).

Content based filtering systems present an automatic approach based on matching user profiles against document representatives. This system automatically generates descriptors of each item's content, and then compares these descriptors to user's profiles, which are descriptors of the user's information need, to determine if the item is relevant to the user. The user's profiles are either supplied by the user, such as in a query, or learned from observing the content of items the user consumes (Herlocker, 2000).

Collaborative filtering system (sometimes known as recommendation system or social filtering), has been developed to address areas where content-based filtering is weak. Collaborative filtering systems are different from traditional computerized information filtering systems in that they do not require computerized understanding or recognition of content. In a collaborative filtering system, items are filtered based on user evaluations of those items instead of the content of those items (Herlocker, 2000). Collaborative filtering is also defined as "a process by which information on the preferences and actions of a group of users is tracked by a system which then, based on observed patterns, tries to make useful recommendation to individual users" (Kumar et al, 1998). It is a social approach based on matching user profile against other user profiles. It uses

other users' judgments that have a similar profile and matches the profile against other profiles and choose the information in the nearest one (Ferreira and Silva, 2001)

Content based and collaborative filtering are intelligent agent based implementations i.e. intelligent software agents that locate and retrieve information with respect to users' preferences (Billsus and Pazzani, 1998).

1.1.2. Software Agents

There is no agreed upon definition of software agents. Different researchers give different definitions of what they call an agent. Agents are sometimes considered as delegates (Meas, 1997) i.e. assistants and at other times viewed as programming objects (Shoham, 1997). According to Shoham (1997) "an agent can be thought of as an extension of the object oriented programming approach where the objects are typically somewhat autonomous and flexibly goal directed."

Lieberman (1994), on the other hand, defined software agent as "a program that can be considered by the user to be acting as an assistant or helper, rather than a tool in the manner of a conventional direct-manipulation interface". He added that an agent should display some [but perhaps not all] of the characters that we associate with human intelligence: learning, inference, adaptability, independence, creativity etc.

Agents are classified in different ways by different researchers. The following is a practical classification of agent applications (based in some cases on what the agents are, and in others on the roles they perform) (Stenmark, 1998; Nwana, 1996).

Interface agents: These are used to decrease the complexity of the more and more sophisticated and overloaded information systems available. They may add speech and natural language understanding to otherwise dumb interfaces, or add presentation ability to systems.

System agents: Such agents run as integrated parts of operating systems or network protocol devices. They help managing complex distributed computing environments by doing hardware inventory, interpreting network events, managing backup and storage devices, and performing virus detection. These agents by their very nature do not primarily work with end-user information.

Mobile Agents: software processes capable of roaming wide area networks (WANs) such as the World Wide Web (WWW), interacting with foreign hosts, performing tasks on behalf of their owners and returning 'home' having performed the duties set for them. These duties may range from making a flight reservation to managing a telecommunications network.

Information Agents: proactive, dynamic, adaptive and cooperative WWW information managers which perform the role of managing, manipulating or collating information from many distributed sources.

Navigation agents: are used to navigate through external and internal networks, remembering short-cuts, pre-load caching information, automatically book-marking interesting sites.

Monitoring agents: these agents provide the user with information when particular events occur, such as information being updated, moved, or erased.

Filtering (content based) agents: are used to reduce information overload by removing unwanted data, i.e. data that does not match the user's profile, from the input stream.

Recommender/Collaborative filtering agents: are usually collaborative; they need many profiles to be available before an accurate recommendation can be made. According to Griffith and O'Riordan (2000), "Collaborative filtering techniques recommend items to users based on the rating of items received from other users with similar tastes to the current user." It is based on the observation that people are good editors and filters of their friends. ***This agent is the agent that is of interest to this research.***

1.1.3. ILRI SDI Service

ILRI SDI service is a computerized SDI/alerting service provided by the ILRI (International Livestock Research Institute) information service. According to my (the researcher) informant Ato Getachew, head of the ILRI SDI service department, the service is designed to keep scientists engaged in research on livestock production and related topic informed of new developments in their particular fields of interest by regularly scanning the worldwide animal agriculture literature. He added that the mission of ILRI's information service is to ensure that ILRI researchers and NARS (National Agricultural Research System) partners are provided with information support in a comprehensive and responsive manner.

1.2. Statement of the Problem and Justification

SDI systems need to filter the huge available information according to users' interest. This will enable the SDI service providers to fulfill their objective of keeping users of a system informed

of the development in their respective fields of interest. It will also allow the users find what they want in a large set of information,

The process of information filtering needs maintaining profiles which are representations of the user interest/information needs (descriptors or keywords). User information needs are dynamic; they change form time to time. Hence there is a need for a system that can manage and maintain user interest profiles well. This need arises because information filtering system must be designed as system that can provide information that matches user needs in a consistent and timely manner (Dawit, 1998). The system must also accommodate change in user needs and adapt to these changes.

It is therefore important that SDI systems, like the ILRI SDI service, which is the case SDI system considered for the purpose of this research, build tools that help the service providers manage and maintain user interest profiles better. This enables the SDI systems to serve the information needs of their users better.

The major limitation of the existing ILRI SDI system, as observed by the present researcher and also pointed out by other workers (Dawit ,1998; Abebe, 2001), is that the process of filtering and updating profiles is very time consuming and inefficient. The reasons for this as stated by Abebe (2001) are:

1. Feedback from users, which is a very essential part in keeping user preferences up-to-date, is rarely obtained.
2. There is no simple mechanism to chase the users provide their feedback on the SDI outputs they receive.

3. The task of analyzing the limited feedback received is restricted to only keeping statistics on MS Access database, which records the number of items sent, feedback statistics (number of items rated as relevant or non-relevant), number of item for which a copy was requested, and the precision ratio calculated, etc.

4. Profile must be optimized and unnecessary terms must be eliminated from the user profile because the outputs contain a huge list of items.

5. Due to the shift of the ILRI bibliographic database on which SDI outputs is produced, from the CDS/ISIS based database to INMAGIC database management software, the matching is not now done automatically. Intermediaries cut the keyboards from the profile file and paste it to the INMAGIC database query form to search and to the matching one-by-one for each user. The filtering activity is now, as a result, time consuming and tasking on the part of the system.

It is therefore recommended that the ILRI SDI service keeps up with the current technological development as information filtering agents (Dawit, 1998). This will help to cope up with the above limitations and be able to maintain the user profiles in far better way. The system will, as a result, provide successful and efficient service.

It was with this basic understanding that Dawit (1998) and Abebe (2001) explored the possibility of using agents for managing and maintaining user interest profiles. Dawit (1998) and Abebe (2001) showed that software agents can be effectively used for information filtering. As mentioned earlier, there are two major classes of information filtering agents: content based filtering and collaborative filtering. Their researches were a content based filtering type. They also used the ILRI SDI service data for their research. Dawit (1998) attempted to improve the

user profile management by prototyping New Interest Tracking Agents. It is a user side agent that "autonomously traces user's new interests and carries out the routine process of informing the SDI server of these interests" (Dawit, 1998). Abebe (2001) tried to model Delivery Agent and Feedback Handler Agent.

The limitation with content based filtering systems is that they only deal with automatic creation of representatives of documents (Ferreira and Silva, 2001). They attempt to find articles of interest to users often based on some scoring functions to evaluate features of the document and return documents with highest score (Maltz, 1994). These limitations, however, were not known by the previous workers, i.e., Dawit (1998) and Ababe (2001).

I want to investigate the possibility of applying collaborative filtering to SDI services. This is because collaborative based filtering systems involve humans in the information filtering process (Herlocker, 2000). Human involvement is important since human beings are better at evaluating documents than a computed function (Maltz 1994).

In addition to adding human subjectivity to the filtering process, collaborative filtering systems, is also thought to solve the problem of relevance feedback to some extent. There is now substantial evidence that the explicit rating of users of the documents they received as relevant or non-relevant helps a lot in the updating of their profiles accordingly (Whitehall, 1979; Kemp, 1979; Ferreira and Silva, 2001).

Collaborative filtering systems as such have the potential to address the issue of user profile management by making use of the relevance feedback of other people. This is observed from the fact that collaborative filtering works by automatically retrieving and filtering documents by

considering the recommendations or feedbacks given by other users (Vel and Nesbitt, 1998). The goal of collaborative filtering is, hence, to predict the preferences of one user, referred to as the active user, based on the preferences of a group of users of similar interest (neighborhood) (Goldberg et al, 2000). For example, given the active users ratings for several documents and a database of other users' ratings, the system predicts how the active user would rate unseen documents (Pennock and Horvitz, 2000).

The idea here is to make agents learn the similarities between user profiles and then, make recommendation. There are two basic assumptions:

- In collaborative filtering model if users usually like same type of documents they belong to the same neighborhood (interest group). This implies that the item that is found interesting by a member(s) of the group (neighbors) will also be found interesting by other members of the same group.
- There are some users who give the required feedback among the neighbors. Accordingly the purpose is to use the feedback from such users to update the profiles of those members who don't give feedbacks as much as needed. This is because, collaborative filtering is based on the premise that people looking for information should be able to make use of what others have already found and evaluated (Maltz and Ehrlich, 1995).

It is therefore the purpose of this research to explore to what extent the application of collaborative filtering agents will extend the use of content based filtering agents further in terms of addressing the problem of user profile updating and maintenance.

1.3. *Objective of the Study*

The general and specific objectives of the study are the following.

- ***General Objective***

The general objective of the study is to investigate the possibility of applying a collaborative filtering agent for making document recommendation in SDI systems with particular reference to ILRI SDI service. With this general goal in mind the research will be undertaken to fulfill the following specific objectives.

- ***Specific Objectives***

The specific objectives of the study are:

1. To study issues related to SDI in general and collaborative information filtering in particular.
2. To review the ILRI SDI service.
3. To explore further the use of IF techniques in improving SDI service provision.
4. To outline a model of the SDI profile updating process so as to allow agent based solutions.
5. To develop and test a prototype model using ILRI SDI service as a case area.
6. To forward recommendations for further work.

1.4. Research Methodology

The following methods are utilized in the process of conducting the study.

- **Data Collection techniques**
  - **Literature Review**

Relevant related literatures dealing with SDI services and software agents have been reviewed. Moreover, to learn what others have done in the area and to better understand the problem a comprehensive investigation of available literature on collaborative filtering has been made.

This study has mainly relied on the Internet for current literature on collaborative filtering.

- **Data Collection**

Frequent contacts have been made with the SDI staff of ILRI information service and various data collection methods including informal interviews, observations and review of documents were used to gather information on the overall task of the service provision. Most importantly, the feedbacks or ratings of users of the information system are obtained and used as an input for the experiment.

- **Programming Technique**

MS Access database management system is used to store the necessary database objects. Two tables and four queries are created and used. The major table is the ratings table which holds the ratings given by every user to items they have seen. This database system is chosen for its availability, simplicity and easy of development and use of queries. The programming language

used for coding the collaborative filtering algorithm is Visual Basic (VB). VB is preferred as a programming language due to familiarity of use and its nature of easy connectivity to Access database.

- **Testing Techniques**

The performance of the collaborative filtering algorithm was tested using coverage and accuracy. Accuracy is measured by precision. These evaluation techniques are used since they are appropriate for evaluating SDI systems.

1.5. Scope and Limitation of the Study

This study tried to accomplish three major tasks. The first task was trying to establish the need of agents in the improvement of SDI service provision. Second, it attempted to illustrate how collaborative filtering agent can be applied. And third it tried to demonstrate the possibility of use of collaborative filtering to address the problem at ILRI SDI. Due to the limited time available to the study, the third component has not been fully accomplished. The short time available for the experiment affected the full testing of the system since testing requires uses to review the output of the system and provide feedbacks.

Another major limitation of the research was the continuous power break down in the city. A further limitation of this study was the very slow Internet connectivity of the university that affected the research in many ways (for example, inability to download important files and software from the Internet).

1.6. Organization of the Thesis

This paper is organized in five chapters. Chapter one introduces the basic background of the research. It lays the ground for the need of a service that can reduce the effort of a user to look for and use relevant information. The second chapter provides the overview of SDI service. This chapter also gives a brief description of the ILRI SDI service. Chapter three is a survey of concepts and researches on collaborative filtering. It shows all the necessary components of collaborative filtering. The fourth chapter presents the overall process of the experiment including data preparation, output generation, testing, evaluation and discussion. The final chapter presents the conclusions drawn and gives recommendations for further research on the area of applying agent technology in general to SDI service system.

Chapter Two

SDI: an Overview

2.1. Introduction

In this chapter an overview of a Selective Dissemination of Information service is given. The first section provides definition, the next part deals with origin and development of SDI service and the following sections give purpose of SDI and Steps in SDI. The last part of this chapter covers an overview to the case SDI System

2.2. Definition

SDI is the provision of a notification service tailored to the specific needs of individuals. It has been defined by different scholars in a quite similar ways as follows:

SDI is defined by Pao (1989) as

> *"a service whose primary function is to alert and notify its clients of potentially useful new information on an individualized basis. It produces a continuous and dependable service, which often extends to the supply of actual documents or abstracts which have been screened and filtered by the systems staff."*

According to Leggate (1975), "SDI refers to a computer assisted subscription based personalized service that notifies a user about new literature and data in accordance with his/her statement of interest." It is an information filtering service (Yan and Garcia-Molina, 1994) that keeps

information to selectively flow to the interested user instead of making the user to go after the information.

These definitions confirm that SDI service is basically a personalized service targeted to fulfill individual information needs. Personalization is achieved through screening or filtering of documents based on the individuals' information needs or requirements.

There are two main sets of inputs to the SDI system: user profiles and documents (Altinel and Franklin, 2000). User profile is an expression of the current interest and needs of a client/user, stated in a way that enables it to be used to identify the records of documents that will meet those interests and needs (Kemp, 1979). A document is an item to be filtered.

According to Yan and Garcia-Molina (1994), SDI service should have the following aspects:

1. Allow a rich class of queries as profiles;
2. Be able to evaluate profiles continuously and notify the users as soon as a relevant document arrives; and
3. Efficiently and reliably distribute the documents to subscribers.

2.3. Origin and Development (trends)

The concept of SDI is an old one. Kumar (1978) argued that librarians have been providing SDI service on manual basis for long time. It is during late 1950's that the idea of computer based SDI service was first formulated. According to Kumar (1978), two problems led to this development. These are:

- The volume and variety of literature being published in various fields especially in science and technology become enormous.
- The information officers (librarians) found it difficult to know all the interests of the users being served. Very often the interest kept on changing.

Kumar (1978) believed that due to these problems, the matching of content of documents with the interest of the users to be done manually become very difficult. That was the time when Luhn suggested a machine system for handling a large scale work of matchmaking.

> *"In 1958 Luhn wrote a paper entitled 'A Business Intelligence System' in which he proposed an automatic method to provide current awareness service to scientists and engineers faced with the ever growing volume of literature. Luhn's technique assumed the availability of text in machine readable form and would consist of automatic abstracting and matching of these abstracts against profiles of users, which he called action points. The result of a match would be a notice containing the abstract and relevant information sent to a subscriber" (Kumar, 1978).*

The use of computers in SDI service has, no doubt, added a great deal for the successful operation of the system. As Leggate (1975) noted SDI become one of the successful services.

The earlier computerized method of providing SDI service was the traditional database technique. This technique is not sufficient in aiding users since the vast majority of information is available in an unstructured or semi-structured format (Griffith and O'Riordan, 2000; Yan and Garcia-Molina, 1993).

This insufficiency of database systems has led to the development of information retrieval (IR) and information filtering (IF) techniques. These systems are basically interested in selecting a subset of documents from the document set, which are relevant to a user's interest (Griffith and O'Riordan, 2000).

As a result, recent researches have focused on the use of information filtering and retrieval techniques for user profile modeling and matching (Altinel and Franklin, 2000). The IR based SDI is in some ways limited, given that: one, it only deals with textual documents, and two, it focuses only on effectiveness of profiles rather than the efficiency of filtering (Altinel and Franklin, 2000). Thus agent based information filtering systems are gaining momentum.

SIFT (Yan and Gracia-Molina, 1999), MySDI (Ferreira and Silva, 2001) are some examples of agent based information filtering systems. Koubarakis et al (2001) also used middle agents to match documents with profiles.

2.4. Purpose of SDI

The most important function of any information service is the dissemination of information that will keep its users well informed and up to date (Kumar, 1978). That is why the main objectives of SDI service is mostly being discussed as helping end users find what they want in a large set of information and keeping them up with the latest developments in their area of interest.

As indicated by Leggate (1975) the purpose of SDI is reducing user effort in selecting from a huge set of information available. Yan and Garcia-Molina (1994) argued "to help users cope with information overload; SDI will increasingly become an important tool in information systems." For Yan and Garcia-Molina SDI is a means of dealing with the problem of information overload.

Quite similarly, Whitehall (1979) stated the primary purpose of SDI service as saving the time and effort of users in reviewing information sources and turning up useful items "which will help them to work more effectively or which will alter the direction of their work to some advantage."

Hence the quality of SDI service is to be measured in relation to its purpose, i.e. how much it fulfilled the interest of users and come up to their expectation. Does the service cover all the relevant literature, how much of the relevant literatures are retrieved and how much irrelevant literature are retrieved (Whitehall, 1979)? For Whitehall there are five things that affect the quality of any SDI service. These are: recall (which measures the ability of the system to present all relevant documents), precision (the ability of the system to withhold non-relevant documents), timeliness (how recent is the information), the number of items sent (notifications) to users, and the proportion of items that are already seen by users in the notification list. Users should not be made to receive too much information at the same time. Kemp (1979) also agrees with Whitehall but added cost of running the profile to the list.

2.5. Steps in SDI

There are basically six steps in providing SDI service.

1. *Creating user profiles:* - a profile consists of keywords that collectively characterize the subject interest of an individual (Whitehall, 1979; Kumar, 1978; Kemp, 1979). The aim of creating profiles is to define the topics of interest appropriately so as to be able to translate the interests into machine readable queries (Kumar, 1978). Some writers, like Ferreira and Silva (2001), argue that profiles have two categories: positive and negative. Positive profile is used to retrieve documents of interest while negative profile is used to

avoid non relevant documents. A profile can be directly obtained from users (say by filing forms) or indirectly learned by the system without too much user involvement through the study of user behavior (Ferreira and Silva 2001).

2. *Processing the profile:* - the profiles obtained thus need to be processed and represented mostly using keywords.
3. *Receiving and processing documents:* - the documents received also need to be processed and represented (indexed).
4. *Matching:* - after the profiles and documents are represented, the two are matched in order to filter documents according to the individual needs.
5. *Sending the notification:* - the filtered documents are then sent as notifications to the user either in a print or electronic format via the electronic mail.
6. *Receiving feedback and updating profiles:* - relevance feedback is the important part of the step since it is the only means of updating user profiles and keeping up with changing user needs.

2.6. *ILRI SDI Service: the Case SDI System*

The mission of ILRI's information service is to ensure that ILRI researchers and NARS (National Agricultural Research System) partners are provided with information support in a comprehensive and responsive manner (Abebe, 2001). According to Dawit (1998), "the ILRI information service is an information service unique in tropical Africa that is backed up by a documentation center that include not only an extensive library on livestock and related topics, but also a major collection of non-conventional literature and a variety of computer based information products and services."

that are not important; statement of expected level of coverage, i.e. comprehensive or specific (which dictates whether the search terms are restricted or not to a specific portion(s) of the record).

Information staff will then convert the above into a user profile. Consideration is given to the specific projects that the user is involved as opposed to the description of user interest. The user profile established in this manner will then be run on a sample database. Based on the results the necessary refinement will be undertaken and passes on to testing it on actual database. The user profile is structured using the query language of the database under operation i.e. INMAGIC database management software (W/ro Maria).

After the profile is constructed matching continues. The ILRI SDI service alerts it's users to the most recent world-wide literature in topics of users' interests. The information is drawn from two international databases (AGRIS and CAB International) to which information acquired by ILRI library is added. The ILRI SDI service to individual users is based on statement of interest maintained against individual user (i.e user profile). Therefore, the ILRI SDI service is the result of the matchmaking process of the two profiles the bibliographic database records and the user profile (W/ro, Maria).

The ILRI SDI service then produces outputs. The outputs has two major components:- the first is the ILRI SDI services output list (which consists of the name and address, the profile, total items matching the user's profile, the list consisting of full bibliographic information including abstracts if any and where the hard copy of the documents could be found). The second component is a tear- off sheet consisting of the feedback information required of users mainly consisting of two statements:

- ❖ Did you find this item relevant? [Yes/No/can't says]
- ❖ Do you want a copy of this item? [Yes/No]

The user is expected to fill in his/her response, tear the printout into two and mail back the feedback copy (i.e. the column containing the feedback questions).

Upon receiving feedback, statistical database maintained by the information section will be updated by counting the number of relevant and non-relevant items as judged by the users. The feedback forms will then be forward to the library to supply a photocopy of the articles for users requesting a copy on their feedback (W/ro Maria).

For further detail on SDI services a user can refer to Dawit (1998) and Abebe (2001).

Chapter Three

Collaborative Filtering

3.1. Introduction

In this chapter seven major issues concerning collaborative filtering will be discussed. The chapter starts by giving an overview to the general concept of collaborative filtering. Following, it presents the limitations of collaborative filtering systems. It then moves on to the issue of algorithms. It continues by noting what steps should be followed in evaluating collaborative filtering systems. The next section points out the evaluation measures used to evaluate CF systems. After that, related researches in the area are summarized. Finally, the system architecture is presented.

3.2. Overview of Collaborative Filtering

The term collaborative filtering (CF) was first coined by Goldberg et al in 1992 at Xerox PARC as part of development of the Information Tapestry system (Goldberg et al, 2000; Maltz, 1994). It has been also known by different names: social information filtering (Shardanand and Maes, 1995) and recommendation system (Schafer, Konstan and Riedl, 2000).

Collaborative filtering has been defined by Chau et al (2002) as "a collaboration in which people help one another perform filtering by recording their reactions to documents they read." In collaborative filtering, a user's actions and analyses regarding a particular piece of information are recorded for the benefit of a larger community i.e., potential users of the same information. Members of the community can benefit from others' experience before deciding to consume new

information. In essence, collaborative filtering systems essentially automate the process of "word-of-mouth" recommendations (Shardanand and Maes, 1995).

As indicated by Herlocker (2000), collaborative filtering systems work by collecting human judgments (known as rating) for items in a given domain and matching together people who share the same information needs or the same tastes. He maintained that users of a collaborative filtering system share their analytical judgments and opinions regarding each item that they consume so that other users of the system can better decide on which items to consume. In return a collaborative filtering system provides useful personalized recommendations for interesting items. CF relies on a very simple idea (Lashkari, 1995): if person A correlates strongly with person B in rating a set of items, then it is possible to predict the rating of a new item for A, given B's rating for that item.

As mentioned in chapter one, the goal of collaborative filtering is, hence, to predict the preferences of one user, referred to as the active user, based on the preferences of a group of users of similar interest (Goldberg et al, 2000). For example, given the active users ratings for several documents and a database of other users' ratings, the system predicts how the active user would rate unseen documents. "The key idea is that the active user will prefer those items that like minded people prefer. The effectiveness of any CF systems is ultimately predicted on the underlying assumption that human preferences are correlated" (Pennock and Horvitz, 2000).

Shardanand and Meas (1995) generalized collaborative filtering as a three step process:

1. The system maintains a user profile, a record of the user's interest (positive as well as negative) i.e., user's ratings of specific items.

2. It compares this profile to the profiles of other users, and weighs each profile for its degree of similarity with the user's profile. The metric used to determine similarity can vary.
3. Finally, it considers a set of the most similar profiles, and uses information contained in them to recommend (or advise against) items to the user.

The following diagram depicts these steps:

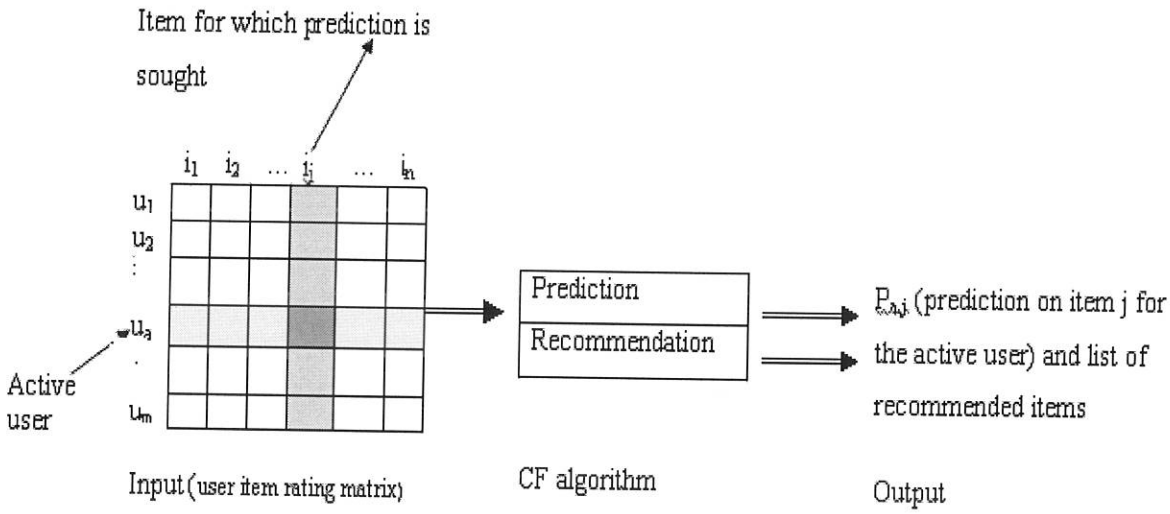

Figure 1: The collaborative filtering process

Collaborative filtering systems can be broadly categorized into *active* and *automated* systems (Herlocker, 2000; Vel and Nesbitt, 1998). In active systems the user must actively identify other users with similar interests and with whom collaboration will be beneficial. The users of the system explicitly forward items of interest to other individuals or groups of people who might be interested. This can place large demands on the users since they have to define the group of collaborators (Vel and Nesbitt, 1998).

On the other hand, automated collaborative filtering (ACF) systems alleviate such demands by automatically generating the collaboration groups (Vel and Nesbitt, 1998). ACF systems automate all procedures except for the collection of user ratings on items. A user of an ACF system rates each item he/she experienced based on how much that item matches his/her information need. These ratings are collected from groups of people, allowing each user to benefit from the ratings of other users in the community. An ACF system uses the ratings of other people to predict the likelihood that an item will prove valuable to the user (Herlocker, 2000).

Simple ACF systems present to a user the average rating of each item of potential interest (Maltz, 1994). This allows the user to discover items that are of popular interest, and avoid items that are of popular dislike. Such ACF systems are not personalized, presenting each user with the same prediction for each item. More advanced ACF systems automatically discover predictive relationships between users, based on common patterns discovered in the ratings. These systems provide every user with personalized predictions that may be different from everybody else's (Herlocker, 2000).

3.3. Limitation of Collaborative Filtering Systems

Even though CF systems have been used fairly successfully in various domains (Melville, Mooney and Nagarajan, 2002) they suffer from four problems:

- **Sparsity: -** Most users do not rate most items and hence the input to the system, i.e., user-item rating matrix is typically very sparse. Therefore the probability of finding a set of users with significantly similar ratings is usually low. This is often the case when systems have a very high item-to-user ratio. This problem is also very significant when the system

is in the initial stage of use (Goldberg *et al.* 1992; Resnick *et al.* 1994; Billsus and Pazzani, 1998).

- **New User Problem:** - To be able to make accurate predictions, the system must first learn the user's preferences from the ratings that the user makes. If the system does not show quick progress, a user may lose patience and stop using the system (Billsus and Pazzani, 1998; Lee, 2001; Melville, Mooney, and Nagarajan, 2002).

- **Recurring Startup Problem:** - New items are added regularly to recommender systems. A system that relies solely on users' preferences to make predictions would not be able to make accurate predictions on these items. This problem is particularly severe with systems that receive new items regularly, such as an online news article recommendation system (Goldberg *et al.* 1992; Resnick *et al.* 1994; Melville, Mooney, and Nagarajan, 2002).

- **Scaling Problem:** - Recommender systems are normally implemented as a centralized web site and may be used by a very large number of users. Predictions need to be made in real time and many predictions may potentially be requested at the same time. The computational complexity of the algorithms needs to scale well with the number of users and items in the system (Lee, 2001; Billsus and Pazzani, 1998).

The potential for collaborative filtering to enhance information filtering tools is great. However, to reach the full potential, it must be combined with content-based information filtering technology. Collaborative filtering by itself performs well predicting items that meet a user's interests or tastes, but is not well-suited to locating information for a specific content information need. In this particular research, however, pure collaborative filtering will be addressed for

reasons mentioned in the statement of the problem and due to the available limited time to integrate both filtering techniques.

3.4. Collaborative Filtering Algorithms

This section presents an algorithmic framework for performing collaborative filtering. The problem of automated collaborative filtering is to predict how well a user will like an item that s/he has not rated given that users ratings for other items and a set of historical ratings for a community of users. A prediction engine collects ratings and uses collaborative filtering technology to provide predictions. An active user provides the prediction engine with a list of items, and the prediction engine returns a list of predicted ratings for those items. Most prediction engines also provide a recommendation mode, where the prediction engine returns the top predicted items for the active user from the database (Herlocker, 2000).

The problem space can be formulated as a matrix of users versus items, with each cell representing a user's rating on a specific item (Griffith and O'Riordan, 2000; Herlocker, 2000). Under this formulation, the problem is to predict the values for specific empty cells (i.e. predict a user's rating for an item). In collaborative filtering, this matrix is generally very sparse, since each user will only have rated a small percentage of the total number of items.

The most prevalent algorithms used in collaborative filtering are what we call the neighborhood-based methods that was first introduced by the GroupLens project (Griffith and O'Riordan, 2000; Herlocker et al., 1999; Melville, Mooney, and Nagarajan, 2002; Herlocker, 2000; Resnick et al., 1994). In neighborhood-based methods, a subset of appropriate users is chosen based on their

similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user. Neighborhood-based methods can be separated into three steps.

1. Weight all users with respect to similarity with the active user.

2. Select a subset of users to use as a set of predictors (possibly for a specific item)

3. Compute a prediction from a weighted combination of selected neighbors' ratings.

Step 1: Calculate User Similarity:

In this step users with similar tastes to the active user are selected. Various techniques can be used to calculate the user correlation. These techniques are among what Breese, Keckerman and Kadie (1998) classified as ***memory-based*** algorithms. "Memory-based algorithms maintain a database of all users' known preferences for all items, and, for each prediction, perform some computation across the entire database" (Breese, Keckerman and Kadie, 1998). The computation that is done through the entire database is similarity computation. Some of the similarity computation algorithms include:

1. *Pearson Correlation*: here a weighted average of deviation from the neighbors' mean is calculated. This approach is used in the original GroupLens system (Resnick et al., 1994). Pearson correlation measures the degree to which a linear relationship exists between two variables and is defined as follows:

$$w_{a,u} = \frac{\sum_{i=1}^{m}\left[\left(r_{a,i} - \overline{r_a}\right)\left(r_{u,i} - \overline{r_u}\right)\right]}{\sqrt{\sum_{i=1}^{m}\left(r_{a,i} - \overline{r_a}\right)^2 \sum_{i=1}^{m}\left(r_{u,i} - \overline{r_u}\right)^2}} \qquad \text{Eq. 1}$$

where $w_{a,u}$ is the similarity weight between the active user and neighbor u, $r_{a,i}$ is the rating given to item i by active user a; $\overline{r_a}$ is the mean rating given by user a; and m is the total number of items.

2. *Constrained Pearson Correlation*: Ringo (Shardanand and Maes, 1995) music recommender expanded upon the original GroupLens algorithm. Ringo claimed better performance by computing similarity weights using a constrained Pearson correlation coefficient, shown in Equation 2. The value 4 was chosen because it was the midpoint of Ringo's seven-point rating scale. Ringo limited membership in a neighborhood by only selecting those neighbors whose correlation was greater than a fixed threshold, with higher thresholds resulting in greater accuracy, but reducing the number of items for which Ringo was able to generate predictions for. To generate predictions Ringo computes a weighted average of ratings from all users in the neighborhood.

$$w_{a,u} = \frac{\sum_{i=1}^{m}\left[\left(r_{a,i} - 4\right)\left(r_{u,i} - 4\right)\right]}{\sqrt{\sum_{i=1}^{m}\left(r_{a,i} - 4\right)^2 \sum_{i=1}^{m}\left(r_{u,i} - 4\right)^2}} \qquad \text{Eq. 2}$$

3. *The Spearman Rank Correlation*: The Spearman rank correlation coefficient is similar to Pearson, but computes a measure of correlation between ranks instead of rating values (Herlocker, 2000). To compute Spearman's correlation, we first covert the user's list of ratings to a list of ranks, where the user's highest rating gets rank 1. Then the computation

is the same as the Pearson correlation, but with ranks substituted for ratings (Equation 3). $K_{a,i}$ represent the rank of the active user's rating of item i. $k_{u,i}$, represents the rank of neighbor u's rating for item i.

$$w_{a,u} = \frac{\sum_{i=1}^{m} [(k_{a,i} - \overline{k_a})(k_{u,i} - \overline{k_u})]}{\sqrt{\sum_{i=1}^{m} (k_{a,i} - \overline{k_a})^2 \sum_{i=1}^{m} (k_{u,i} - \overline{k_u})^2}} \qquad \text{Eq. 3}$$

4. *The vector similarity*: uses the cosine measure between the user vectors to calculate correlation. The formulation according to Breese, Keckerman and Kadie (1998) is:

$$w_{a,u} = \sum_{i=1}^{m} \frac{r_{a,i}}{\sqrt{\sum_{k \in I_a} r_{a,k}^2}} \frac{r_{u,i}}{\sqrt{\sum_{k \in I_u} r_{u,k}^2}} \qquad \text{Eq. 4}$$

where I_a is the set of items which user a rated. The squared terms in the denominator serve to normalize ratings.

5. *Mean-squared difference*: is another alternative that was used in the Ringo music recommender (Herlocker, 2000). Mean-squared difference (Equation 5) gives more emphasis to large differences between user ratings than small differences.

$$d = \frac{\sum_{i=1}^{m} (r_{a,i} - r_{u,i})^2}{m} \qquad \text{Eq. 5}$$

Other similarity measures include the entropy-based uncertainty measure (Griffith and O'Riordan, 2000; Herlocker, 2000). The measure of association based on entropy uses

conditional probability techniques to measure the reduction in entropy of the active user's ratings that results from knowing another user's ratings (Herlocker, 2000).

Herlocker (2000) found that the mean-squared difference algorithm, introduced in the Ringo system, did not perform well compared to Pearson correlation. The vector similarity measure has been shown to be successful in information retrieval; however Breese et al has found that vector similarity does not perform as well as Pearson correlation in collaborative filtering (Breese, Keckerman and Kadie, 1998). Algorithms using Spearman correlation perform worse or the same as comparable algorithms using Pearson correlation. Furthermore, computation of Spearman correlation is much more compute-intensive, due to the additional pass through the ratings necessary to compute the ranks (Griffith and O'Riordan, 2000; Herlocker, 2000). Due to these reasons, Pearson correlation is used as a similarity computation technique for this research.

Step 2: Select Neighborhood

After having assigned similarity weights to users in the database, the next step is to determine which other users' data will be used in the computation of a prediction for the active user. It is useful, both for accuracy and performance, to select a subset of users (the neighborhood) to use in computing a prediction instead of the entire user database (Herlocker, 2000). The system must select a subset of the community to use as neighbors at prediction computation time in order to guarantee acceptable response time. Furthermore, many of the members of the community do not have similar tastes to the active user, so using them as predictors will only increase the error of the prediction.

Another consideration in selecting neighborhoods suggested by Breese, Keckerman and Kadie (1998) is that high correlates (such as those with correlations greater than 0.5) can be exceptionally more valuable as predictors than those with lower correlations can. Two techniques, correlation-thresholding and best-n-neighbors, have been used to determine how many neighbors to select (Griffith and O'Riordan, 2000).

1. *Correlation thresholding*: - where all neighbors with absolute correlations greater than a specified threshold are selected. Selecting a high threshold means that only good correlates will be selected thereby giving more accurate predictions. Sometimes there may be very few neighbors who have such high correlation with the active user and as a result it may not be possible to generate meaningful predictions for some items (Griffith and O'Riordan, 2000). The first technique is used by Shardanand and Maes (1995).
2. *Best-n Correlations*: - where the best n correlates are picked. Picking a large value of n may result in too much noise for those with high correlates whereas picking a small n can cause poor predictions for users with low correlates (Griffith and O'Riordan, 2000).

This research uses best n-neighbors or correlates to select the neighborhood. This technique is recommended by Herlocker (2000).

Step 3: Generate a Prediction

Once the neighborhood has been selected, the ratings from those neighbors are combined to compute a prediction. The basic way to combine all the neighbors' ratings into a prediction is to compute an average of the ratings. The averaging technique has been used in all published

work using neighborhood-based ACF algorithms (Melville, Mooney, and Nagarajan, 2002; Herlocker et al., 1999).

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} \left[\left(r_{u,i} - \bar{r}_u \right) * w_{a,u} \right]}{\sum_{u=1}^{n} w_{a,u}} \qquad \text{Eq. 6}$$

where $P_{a,i}$ is the prediction for the active user a for item i; $W_{a,u}$ is the similarity between users a and u; and n is the number of users in the neighborhood.

Alternative techniques of performing collaborative filtering are Bayesian networks (Breese, Keckerman and Kadie, 1998), singular value decomposition with neural net classification (Griffith and O'Riordan, 2000), and induction rule learning (Basu, Hirsh and Cohen, 1998). Breese, Keckerman and Kadie (1998) called this group of algorithms ***model based*** algorithms. Model-based algorithms first compile the users' preferences into a descriptive model of users, items, and/or ratings; i.e., learns a user model and then make prediction using the model. Model-based systems are based on a compact model inferred from the data (Breese, Keckerman and Kadie, 1998).

Pennock and Horvitz (2000) suggested a third class of collaborative filtering method which they called ***personality diagnosis*** (PD). They said that their method can be seen as a hybrid between memory-based and model-based approaches. All data is maintained throughout the process, new data can be added incrementally, and predictions have a meaningful probabilistic semantics. Personality Diagnosis (PD), is an approach based on computing the probability that a new user is of an underlying "personality type," and that user preferences are a manifestation of this

2. Identify the specific tasks towards those goals that the system will enable

3. Identify system-level metrics and perform system evaluation

3.5.1. Identify High Level Goals

Before measuring the performance of an information filtering system, Herlocker (2000) said, we must determine exactly the goals of the system as well as the exact tasks the users will be performing with the system.

Information filtering systems are not valuable by themselves. Rather they are valuable because they help people to perform tasks better than those people could without assistance from the filtering system. Therefore, at the highest level, the goal of ILRI SDI service is to improve the effectiveness of an existing user task.

If people are currently engaged in the performance of a task, then it is not the part of the information-filtering-system builder to justify that task. Herlocker (2000) argued that the valuable contribution that can be made is to improve significantly the efficiency, quality, or speed of such a task.

3.5.2. Identify Specific Tasks

Having specified what the high-level goals are, the next step is to specify specific tasks that the users will perform, aided by the information-filtering system. These tasks will describe explicitly the nature of interaction between the user and the system (Herlocker, 2000). The choice of the appropriate metric to use in evaluating a system will depend on the specific tasks that are identified for the system.

The appropriate tasks will depend greatly on the high-level goals of the users.

Herlocker (2000) presented 6 different representative tasks that a user of an ACF system might have. He said that these tasks will illustrate the competing features of different metrics.

1. A user wants to locate a single item whose value exceeds a threshold. For example, a common task would be to locate a single "decent" movie to watch, or a book to read next.

2. A user is about to make a selection decision that has significant cost, and wants to know what the "best" option is. For example, a selection between many different health plans could have significant future consequences on a person. They are going to want to make the best possible selection.

3. A user has a fixed amount of time or resources, and wants to see as many of the most valuable items as possible within that restriction. Therefore, the user will be interested in the top n items, where n depends on the amount of time the user has. For example, consider news articles. People generally have a fixed amount of time to spend reading news. In that time, they would like to see the news articles that are most likely to be interesting.

4. A user wants to gain or maintain awareness within a specific content area. Awareness in this context means knowing about all relevant events or all events above a given level of interest to the user. This is a type of task that a user of ILRI SDI service is engaged in. Because, as pointed out earlier, the most important function of an SDI service is the dissemination of information that will keep its users well informed and up to date.

5. A user wants to examine a stream of information in a given order, consuming items of value and skipping over items that are not interesting or valuable. For example, in Usenet bulletin boards, some readers frequently examine the subject line of every article posted to a group. If a subject appears interesting, the entire article is retrieved and read.

6. A user has a single item and wants to know if the item is worth consuming. For example, a user may see an advertisement for a new book and wants to know if it is worth reading. These tasks illustrate several very different manners in which users will interact with ACF systems.

3.5.3. Performing System-Level Analysis

System-level evaluation is performed in cases where researchers can identify measurable indicators of the system that will significantly correlate with the effectiveness of a system independent of the user interaction. Researchers who use system-level evaluation assume that differences in the given indicators will result in better task performance given any reasonable user interface (Herlocker, 2000).

System-level evaluation has been the most prevalent form of evaluation in information filtering, because it offers inexpensive, easily repeatable analysis (Herlocker, 2000). The data are collected from users once, and then many different systems can be evaluated on the collected data without further expensive user sessions.

3.6. Evaluation Metrics

For measuring the performance of recommender algorithms measures originating from statistics, machine learning and information retrieval are used. Given the goal of collaborative filtering

systems, helping users more effectively identify the content they want, the utility of the system is defined to include two dimensions: *coverage* and *accuracy* (Sarwar et al., 1998).

Coverage is a measure of the percentage of items for which a recommendation system can provide recommendations (Griffith and O'Riordan, 2000). A low coverage value indicates that the user must either forego a large number of items, or evaluate them based on criteria other than recommendations. A high coverage value indicates that the recommendation system provides assistance in selecting among most of the items. Coverage is usually computed as a percentage of items for which the system was able to provide a recommendation (Geyer-Schulz and Hahsler, 2002).

$$Coverage = \frac{\text{number of items recommended}}{\text{total number of items}} \qquad \text{Eq. 7}$$

Herlocker (2000) held that coverage may be appropriate for certain ranking-based tasks (potentially appropriate for user task types 1, 2, and 3 described in section 3.5.2.). However, he said, for tasks in which a user can request a prediction for any item in the database (tasks 5 and 6), not be able to produce a prediction is generally inappropriate. In any case, coverage should be reported, and system accuracy should only be compared on items for which both systems can produce predictions (Herlocker, 2000).

Accuracy is a measure of the correctness of the recommendations generated by the system (Griffith and O'Riordan, 2000). It is the fraction of correct recommendations to total possible recommendations. The metrics for evaluating the accuracy of a prediction algorithm can be divided into two main categories: *statistical accuracy metrics* and *decision-support metrics* (Herlocker et al., 1999). Statistical accuracy metrics evaluate the accuracy of a predictor by

comparing predicted values with user-provided values. Decision-support accuracy measures how well predictions help users select *high-quality* items.

Statistical accuracy metrics: - statistical accuracy measures the closeness between the numerical recommendations provided by the system and the numerical ratings entered by the user for the same items (Sarwar et al., 1998). Common metrics used include:

- *Mean Absolute Error* (MAE*) and Related Measures*

Mean absolute error measures the average absolute deviation between a predicted rating and the user's true rating. It has been used to evaluate ACF systems in several cases (Herlocker, 2000).

$$\text{MAE} \quad \overline{|E|} = \frac{\sum_{j=1}^{N} |p_i - r_i|}{N} \qquad \text{Eq. 8}$$

where $|p_i - r_i|$ is the absolute error of each component and N is total number of items we produce recommendations for.

With mean absolute error, the error from every item in the test set is treated equally. According to Herlocker (2000), this makes mean absolute error most appropriate for Type 5 and Type 6 user tasks, where a user is requesting predictions for all items in the information stream, or it is not known which items for which predictions will be requested.

As per Herlocker (2000), MAE may be less appropriate for tasks where a ranked result is returned to the user, who then only views items at the top of the ranking, such as types 1-4. However, he maintained that the mean absolute error metric should not be discounted as a

potential metric for ranking-based tasks. Intuitively, it seems clear that as mean absolute error decreases, all other metrics must eventually show improvements. There are two other advantages to mean absolute error. First, the mechanics of the computation are simple and easily recognized by all. Second, mean absolute error has well studied statistical properties that provide a means for testing the significance of difference between the mean absolute errors of two systems.

Two related measures are the mean squared error and the root mean squared error. These variations square the error before summing it. The result is more emphasis on large errors. For example, an error of one point increases the sum by one, but an error of two points increases the sum by four.

- *Precision and Recall*

Precision and recall are the most popular metrics for evaluating information retrieval systems (Sarwar et al., 1998). Precision and recall are computed from a contingency table, such as the one shown in table 1 below. The item set must be separated into two classes – relevant or not relevant. Recall measures the ability of the system to present all relevant documents. Precision, on the other hand, measures the ability of the system to withhold non-relevant documents (Geyer-Schulz and Hahsler, 2002).

	Selected	Not Selected	Total
Relevant	Nrs	Nrn	Nr
Irrelevant	Nis	Nin	Ni
Total	Ns	Nn	N

Table 1. Contingency table showing the categorization of items in the document set.

If an item meets an information need, then it is a successful recommendation (i.e. relevant). If we measure how likely the system is to return relevant documents, then we are measuring how likely the system meets the user's information need.

Likewise, we need to separate the item set into the set that was returned to the user (selected), and the set that was not. We assume that the user will consider all items that are retrieved. *Precision* is defined as the ratio of relevant documents selected to number of documents selected, shown in Equation 9.

$$P = \frac{N_{rs}}{N_s} \qquad \text{Eq. 9}$$

Precision represents the probability that a selected document is relevant. *Recall* is defined as the ratio of relevant documents selected to total number of relevant documents available. Recall represents the probability that a relevant document will be selected.

$$R = \frac{N_{rs}}{N_r} \qquad \text{Eq. 10}$$

Precision and recall depend on the separation of relevant and non-relevant items. The definition of "relevance" and the proper way to compute has been a significant source of argument within the field of information retrieval (Geyer-Schulz and Hahsler, 2002). Most information retrieval evaluation has focused on an objective version of relevance, where relevance is defined with respect to the query, and is independent of user. Teams of experts can compare documents to queries and determine which documents are relevant to which queries. However, objective relevance makes no sense in automated collaborative filtering. ACF is recommending items

- *Receiver Operating Characteristic* (ROC)-curve

The ROC model attempts to measure the extent to which an information filtering system can successfully distinguish between signal (relevance) and noise. The ROC model assumes that the information filtering system will assign a predicted level of relevance to every potential item (Herlocker, 2000). The ROC-curve is a plot of the systems *sensitivity* (probability of detection, true positive rate) by its 1-*specifity* (probability of false alarm, 1- true negative rate). ROC sensitivity ranges from 0 to 1, where 1 is ideal and 0.5 is random (Melville, Mooney, and Nagarajan, 2002). ROC sensitivity is a measure of the diagnostic power of a filtering system. Operationally, it is the area under the receiver operating characteristic (ROC) curve—a curve that plots the sensitivity and specificity of the test. Sensitivity refers to the probability of a randomly selected good item being accepted by the filter. Specificity is the probability of a randomly selected bad item being rejected by the filter. The ROC sensitivity measure is an indication of how effectively the system can steer people towards highly-rated items and away from low-rated ones (Sarwar et al, 1998).

The ROC area measure is most appropriate for type 4 tasks (where a user wants to gain or maintain awareness within a specific content area), and to a lesser extent, type 3 tasks (where a user has a fixed amount of time or resources, and wants to see as many of the most valuable items as possible within that restriction). In type 4 tasks, users have a binary concept of relevance, which matches with ROC area. Furthermore a type 4 user is interested in both recall and fallout, wanting to see the largest percentage of relevant items with the least amount of noise. ROC area is less appropriate for type 3 users, since type 3 users want the current item in the ranking to be

more relevant than all later rankings and ROC does not guarantee this within the binary class of relevance (Herlocker, 2000).

3.7. Related Research

Collaborative filtering is gaining popularity as a research topic, and many groups are currently working to develop new strategies for it. The following are to mention but a few among the variety of collaborative filters systems which have been designed and deployed.

Tapestry (Goldberg et al., 1992) is one of the earliest implementations of collaborative filtering systems. It is a project where the concept of collaborative filtering originated. The Tapestry system relied on each user to identify like-minded users manually. This system relied on the explicit opinions of people from a close-knit community, such as an office workgroup. While this system has contributed significantly to collaborative filtering it suffers from the following drawbacks: lack of privacy, users know from whom they are getting recommendations, it requires explicit user interaction, and the formulation of the collaborative relationships still remains the task of the user. Collaborative filtering system for large communities can not depend on each person knowing the others. Later on several ratings-based automated collaborative filtering systems were developed.

The early news article recommendation system, GroupLens (Resnick et al., 1994), is a filtering system that combines collaboration with user profiles. The purpose of GroupLens is to increase the value of time spent reading electronic forums (specifically Usenet News). In GroupLens, communities of users rank the articles they read on a numerical scale. The GroupLens system then finds correlations between the ratings users have given the articles. Essentially, the user

profile consists of the ratings that he/she has given to the articles he/she has read. When user A wishes to filter new articles of information, the ratings other users have given those new articles are combined to form a recommendation for A on how interesting the new articles will be for A. The ratings from other users are combined by weighting each user's rating in proportion to how well his user-profile correlates with A's. The goal of the system is to identify a peer group of users whose interests are similar to A's, and then to use their opinions of new articles to predict whether A will like the articles.

The GroupLens Usenet filtering system was the first to provide automated collaborative filtering (Konstan et al., 1997; Miller, et al., 1999). Since then, the GroupLens Research Project at the University of Minnesota has conducted further research in ACF systems, covering areas such as matrix storage methods for ACF, integration of content-based filterbots to improve prediction quality (Good, et al., 1999; Sarwar et al., 1998), combining independent filtering agents with ACF, empirical analysis of prediction algorithms (Herlocker et al., 1999), reducing dimensionality with SVD, and explanation of ACF recommendations (Herlocker, Konstan, and Riedl, 2000).

Several other similar systems were developed around the same time as the GroupLens Usenet system, including the Ringo (Shardanand and Maes, 1995). Ringo is a collaborative filtering system which makes personalized music recommendations. People describe their listening pleasures to the system by rating some music. These ratings constitute the person's profile. This profile changes over time as the user rates more artists. Ringo uses these profiles to generate advice to individual users. Ringo compares user profiles to determine which users have similar taste (they like the same albums and dislike the same albums). Once similar users have been

Hence collaborative filtering clients are programs (e.g., web browsers) that present the user with an interface for browsing documents. These client applications communicate with document servers (e.g. databases) to retrieve documents for the user. Clients also communicate with collaborative filtering engine server through its API. Calls are provided to enter ratings for documents and to request recommendations for documents.

This study deals with only the Interface to the recommendation engine due to the shortage of time available to consider all components.

Chapter Four

Experiment

4.1. Introduction

This chapter presents the research experiment. It first deals with how the data is collected. It then goes on explaining how the experiment is carried out. Finally, the result of the experiment and discussion are provided.

4.2. Data

The data that is used for this experiment is obtained from ILRI SDI service. It consists of 2783 ratings (records) from 39 users. The ratings are in three point scale with one representing negative (not relevant), two representing can not say and three representing positive (relevant). That is, for example user number 010received a specific document (say document number 45367) and said that it is relevant, not relevant or cannot say. This data is prepared from user feedbacks which have the following format:

Feedback from User ID 010

1. "Campi, A.J." "Amalgamating the free market and traditional nomadic society: sustainable economic development for Mongolia" "Akiner, S. (ed.); Tideman, S. (ed.); Hay, J. (ed.)" "Sustainable development in Central Asia" "Central Asia Research Forum Series" "p. 104-117" "2 ref." "Curzon Press Ltd" "Richmond (UK)" "1998"

"MONGOLIA; NOMADISM; PASTORAL SOCIETY; ENVIRONMENTAL MANAGEMENT; ECONOMIC DEVELOPMENT; ECONOMIC POLICY; CULTURAL VALUES; MARKET ECONOMIES; SUSTAINABILITY" "A discussion is presented on the applicability of the economic development models designed in the West for Mongolia. It suggests that they would only meet with limited success because they did not consider the country's environmentally-attuned, nomadic society and value system as key factors. The paper concludes that over the next few years Mongolia should try to evolve into a new society which sill preserves its nomadic traditions and protects the ecology which nurtured it over the centuries, and that this will be much more successful

Did you find this item relevant? [Yes]

A feedback data from 5 users is obtained to test the performance of the system.

4.2.1. The Data preparation Process

The data was first prepared with the assumption that DBLens software, which is a collaborative filtering engine, will be used. DBLens was and still (June 28, 2003) is the only available collaborative filtering software to the knowledge of the researcher. That was the reason for the choice to use the software at that time. The software accepts a numerical data of the form (user_id, item_id and rating).

DBLens is an oracle based toolkit which works on UNIX machines. Since it needs a UNIX machine and it is mentioned in its readme file that it works on any UNIX machines, the Red Hat Linux (version 8) was found and installed on Intel architecture, which is the only available option.

While looking for Oracle, the researcher began preparing the data in a format that would be used as an input to the software. The process of preparation was manual and hence time consuming. Every item forwarded to a particular user (like the example displayed above) needs to be searched in the bibliographic database obtained from ILRI to assign to it an Item ID. After an item ID was assigned the data was placed on paper with the following format:

(010, 45367, 3)
(010, 26072, 1)
(267, 26072, 1)
(267, 50148, 1)
(73, 5398, 3)
(83, 74184, 1)
(520, 54214, 2)
(520, 54223, 1)
(539, 50026, 3)

where the first entry stands for the user ID, the second for item ID and the third for a rating given by the user to that specific item.

This process is done for all feedback copies obtained. Finally there were 2783 feedbacks records for 39 users.

These records are inserted to an SQL server database using the insert into command after the database and the necessary table is created.

```
CREATE DATABASE ILRISDI
CREATE TABLE ratings (
        user_id int NOT NULL,
        item_id int NOT NULL,
```

```
        rating int NULL
    CONSTRAINT ratings_key PRIMARY KEY (user_id, item_id)
    )
    GO

--example of insert rating
    insert into ratings
            (user_id,item_id,rating)
    values (010, 45367, 3)
     insert into ratings
            (user_id,item_id,rating)
    values (267, 26072, 1)
```

The SQL command was preferred at that time because it was thought that Oracle uses the same command and it would be easy to reuse the code when it is found.

The researcher was still looking but couldn't find the Oracle CDs i.e., oracle that run on Intel Linux. The only option left was to look for the software for download from the Internet. Oracle for Intel Linux was found to be freely available to download from otn.oracle.com. There are three CDs to be downloaded and the overall size of the CDs is about 1.5 Giga Bytes. Nevertheless, to download one of the CDs on the proxy network of AAU (Addis Ababa University) requires 150 hours, which is impractical. The researcher thought that the network would be faster during the night and the weekends but there was no real difference. Friends from network administration also tried many times to download the files bypassing proxy servers and using download accelerators. They couldn't succeed because the transfer interrupts always.

Then the researcher tried the download at two different places that were thought to have better connection: UNECA (United Nations Economic Commission for Africa) and ILRI. However, these efforts didn't produce any result as the transfer always gets interrupted.

At this point the researcher gets frustrated. It was at this time that the idea of writing a code (program) comes from supervisors. With their encouragement the researcher starts working on the code using Visual Basic (VB). VB was preferred due to familiarity of use as also mentioned in chapter one. Refer to the appendix A to see the VB code written.

The researcher then thinks that using MS Access database will make the work easier and as a result exported the SQL database to MS Access using the import and export facility of SQL server.

The following is an example of the ratings table in the Access database:

user_id	item_id	rating
10	49843	3
10	50199	3
17	50746	3
17	50760	1
73	5392	2
73	5398	3
83	74184	1
267	8785	3
267	10862	1
520	54223	1
539	50026	3
539	50029	3
543	67116	3
543	67162	3

Table 2: An Example of ratings Table

For the purpose of testing the system the number of records in the original ratings table (2783) was reduced to 2775 by elimination those users who have less than 5 ratings. As a result the number of users is also reduced from 39 to 32. This is done to reduce the dimensionality and the error of the algorithm as users with very few rating will be bad recommenders to any active user. The number of distinct items also reduces from 2358 to 2352.

4.3. *The Collaborative Filtering Model*

This experiment followed the neighborhood approach, which is a three step process: similarity weighting, neighborhood selection and prediction calculation.

The first step in neighborhood prediction algorithms is to weight all users with respect to similarity with the active user. Similarity weights are computed to measure closeness between users. Among the various similarity weighting measures mentioned in the third chapter, Pearson correlation is used for this experiment. This is because, as also mentioned in chapter three, Pearson correlation is the commonly used technique in all neighborhood based researches and it is also tested to perform better than the others. In Pearson correlation a weighted average of deviation from the neighbors' mean is calculated. It measures the degree to which a linear relationship exists between two variables. In order to remind the reader the Pearson correlation equation (equation 1 on page 32) was reproduced here.

$$W_{a,u} = \frac{\sum_{i=1}^{m} \left[\left(r_{a,i} - \overline{r_a} \right) \left(r_{u,i} - \overline{r_u} \right) \right]}{\sqrt{\sum_{i=1}^{m} \left(r_{a,i} - \overline{r_a} \right)^2 \sum_{i=1}^{m} \left(r_{u,i} - \overline{r_u} \right)^2}} \qquad \text{Eq 1}$$

where $w_{a,u}$ is the similarity weight between the active user and neighbor u, $r_{a,i}$ is the rating given to item i by active user a; $\overline{r}_a$ is the mean rating given by user a; and m is the total number of distinct items in the database. Correspondingly, $r_{u,i}$ is the rating given by a user u to item i, $\overline{r}_u$ is the average rating of user u.

The correlation is calculated iteratively over all users excluding the active user. This means that u changes from user 1 up to user n minus one, where n is the total number of users in the database.

For example, if we take user_id 10 as an active user, then user 10 is our a and our u's will be all the other users one by one. That means we first calculate similarity weight between 10 and 17 $(w_{10,17})$ and then similarity between 10 and 22 $(w_{10,22})$ and then similarity between 10 and 44 $(w_{10,44})$ and so on. Note that 10, 17, 22 and 44 are user IDs.

The result of these correlations is a number between -1 and +1. In most of the cases the numbers were between 0.6 and 1 in this research. The following are the examples of the results of the correlation:

0.9220933
0.9130991
0.9899082
0.867824
0.9393129 ...

After these similarity weights are assigned to all users in the database, i.e., correlation between the active user and all other users are calculated, the next step is to determine which other users' data will be used in the computation of a prediction for the active user. It is useful for both accuracy and performance to select a subset of users (the neighborhood) to use in computing prediction instead of the entire user database. Selecting a subset of user community to use as

neighbors at prediction computation time also guarantees acceptable response time when the database is very large. Furthermore, most of the members do not have similar tastes with the active user, so using them as predictors will only increase the error of the prediction.

Best-n neighbors method was employed to select a subset of users (the neighborhood) as predictors. This is because it is suggested in literature. Best-n neighbors are the n neighbors with top W values as compared to other users.

Experiment was performed using two different n's: three best neighbors (which is about 9% of the users as neighbors) and using 8 best neighbors (which is 25% of the users as neighbors). The two numbers (3 and 8) are randomly selected at the beginning. However, other experiments were performed afterwards taking other numbers (2, 4, 5, 6, 7, 9, and 10). Yet no difference is observed in their output. In fact for the numbers 2, 4 and 5 the result is the same as 3; and for the other numbers the same output is gained as that of 8.

Once the neighborhood has been selected, the ratings from those neighbors are combined to compute a prediction. The best way used in all published neighborhood based collaborative filtering techniques is the averaging method. It is the deviation from mean average over all users. The averaging formula (equation 6 on page 36) is also reproduced here for easy reference.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} \left[\left(r_{u,i} - \bar{r}_u \right) * w_{a,u} \right]}{\sum_{u=1}^{n} w_{a,u}}$$

To calculate this formula the correlation values of step one should be ranked in descending manner. And then n other users with top correlation values are selected to be neighbors to the

In the Active User Id box the ID number of the user that is to get the recommendation (the active user) is entered or selected from the drop down list. The No of Best Neighbors box accepts best-n neighbors' value. In the number of best articles box a value is selected or entered specifying the number of articles preferred to be seen. For this research the number of best articles to be displayed for the user is chosen to be 15. This number was randomly chosen but the idea in mind was that if the number to be displayed at a time is large the user will be unable (feel overwhelmed) to see all items and give feedback.

The following figure displays the items predicted for active user 010 (user_id 010) when the number of best neighbors is only three.

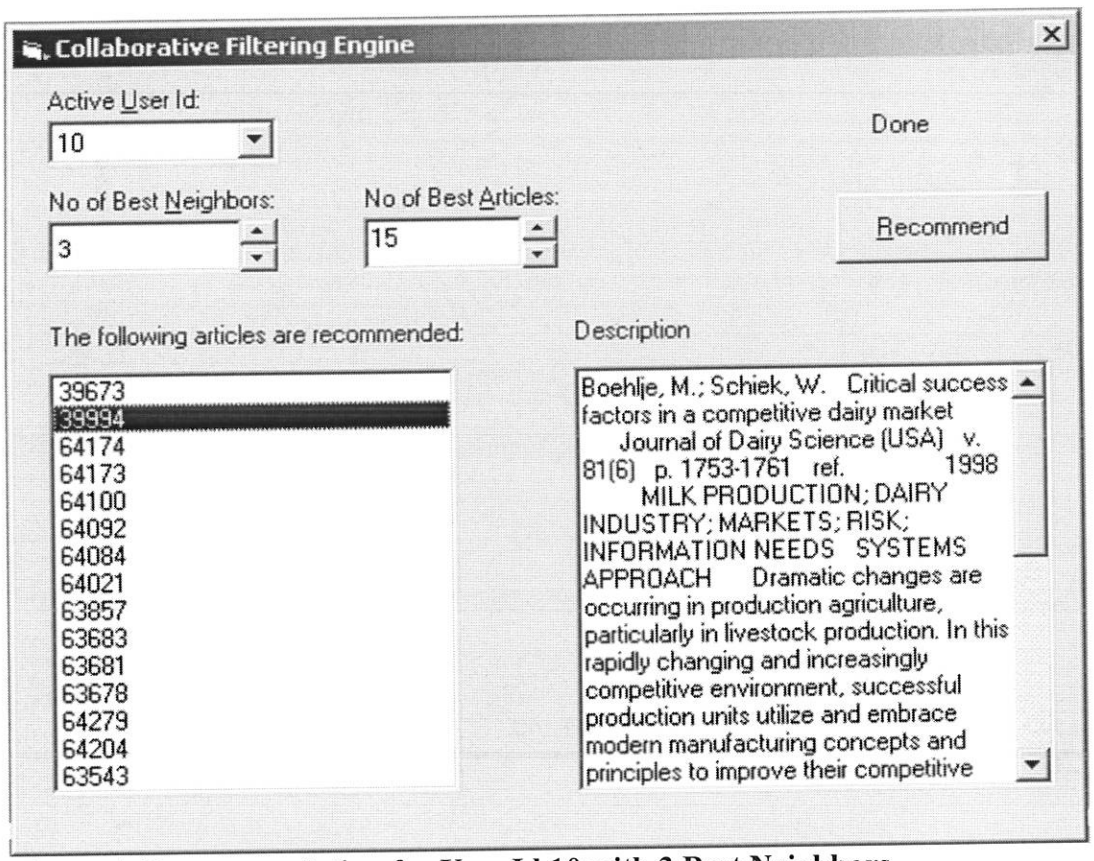

Figure 4: Recommendation for User Id 10 with 3 Best Neighbors

When a user clicks on any article ID its corresponding description (all available bibliographic information including the abstract) is displayed in the Description box simultaneously.

The following figure displays the items predicted for active user 010 when the number of best neighbors used is eight.

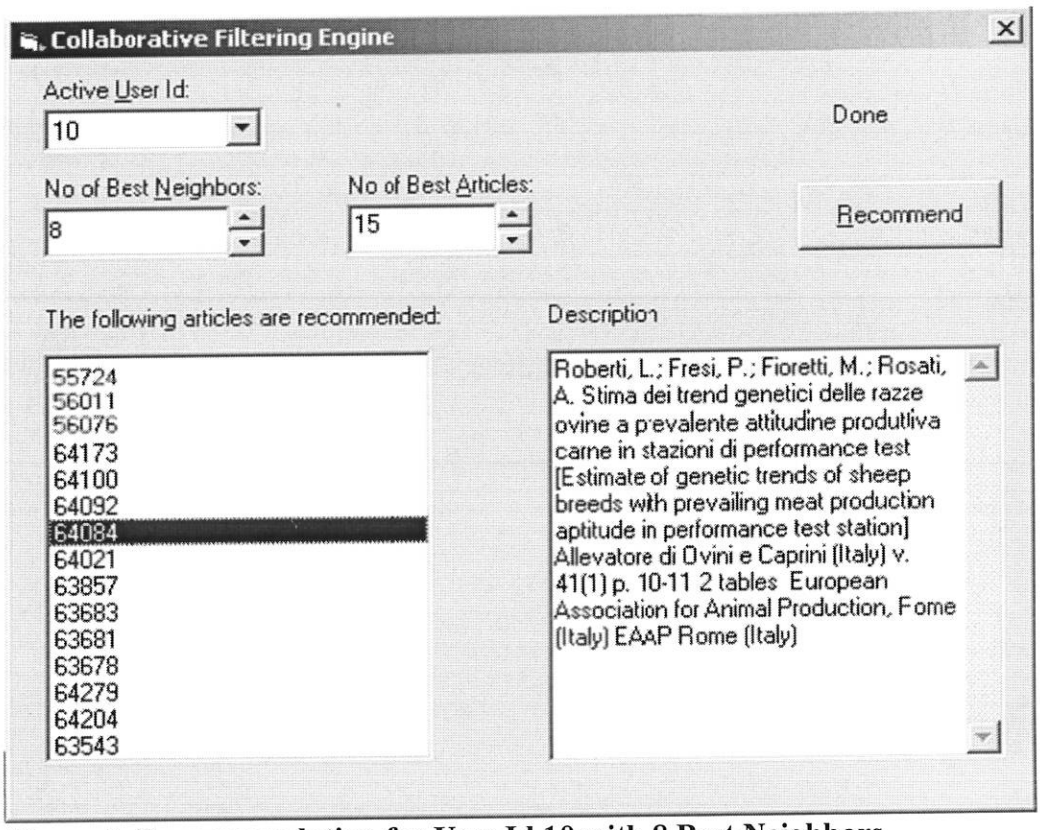

Figure 5: Recommendation for User Id 10 with 8 Best Neighbors

The predicted items are displayed in their order of prediction value (value of $p_{a,i}$). That means the top item is predicted to be of best interest to the active user.

After this output is produced the corresponding items (title and abstract of each recommended item) are copied and sent to the all the 10 users accordingly. See the sample output copy sent in appendix c.

4.5. The System Performance and Analysis

There are two performance measures used in this research: coverage and accuracy as measured by precision. Precision is taken as the only accuracy measure due to the shortage of time. Recall requires a domain expert to sit and evaluate the relevance of all 2352 items for all users (at least 10 users for this test case). This requires so much time of the expert and the willingness to spend that much time voluntarily.

4.5.1. Coverage

Coverage is the measure of the percentage of items for which the collaborative filtering system can provide prediction (refer to chapter three). For this research when all other users are taken as neighbors the coverage is 99%. That is, it has a maximum coverage. However, as the number of neighbors decrease coverage also decreases accordingly. For example when the number of neighbors is three, coverage is nearly only 9%, which is very low. When the neighborhood size is eight the coverage grows to 30% which is still low but much better than when the size is three. This is expected since decreasing the number of neighbors means limiting the size of items to be recommended to items only seen by those users excluding the items seen by the active user. The following figure (figure 6) shows the coverage graph of this experiment. It summarizes the whole discussion.

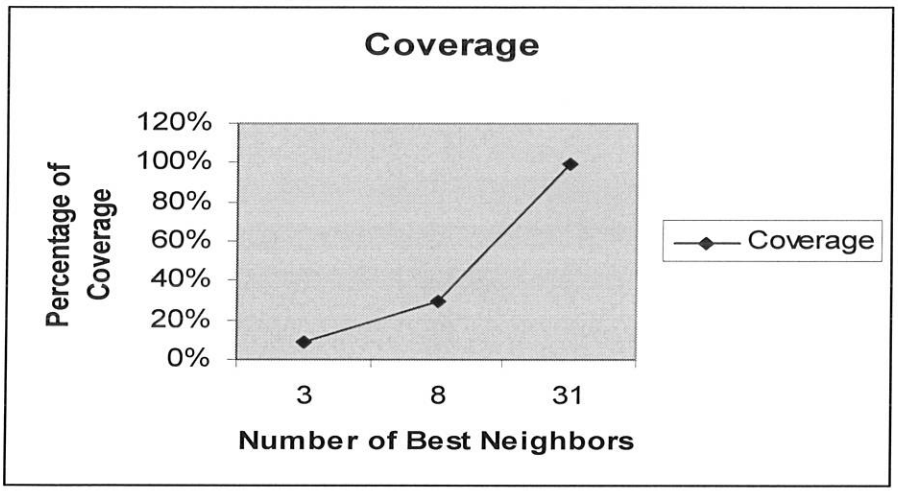

Figure 6: A Graph Showing Number of Best Neighbors by Coverage Levels

This figure shows that coverage rises as the number of best neighbors taken as predictors grow.

4.5.2. Precision

Out of the ten users required to evaluate the performance of the system, only five users responded. One of them (User ID 10) responded by only saying that the output is generally relevant without specifying which items are relevant and which are not. This made the calculation of accuracy for that user impossible. The following table summarizes the precision result obtained from the other four users.

User ID	Precision at 3 Neighbors	Precision at 8 Neighbors	Ratings of a User
153	33.3%	6.7%	14
056	53.3%	6.7%	18
83	60%	33.3%	51
543	73.3%	53.3%	297

Table 3: Accuracy Measure of Precision for two Different Best Neighbor sizes

The result depicted in table 2 shows that small neighboring size performs well in most cases. The result contradicts the idea of Herlocker (2000) which mentions that accuracy of a collaborative

filtering system decreases with the decrease in the number of neighborhood size used for recommendation. Yet, it may also be the case that when the number of items and users is relatively small, small neighborhood size is preferable. The fourth column of table 2 shows the ratings that users have in the database. For example, user Id 153 had only given feedback to 14 of the 2352 items available in the database, whereas 543 gave 297 ratings. The following is a graphical representation of the table which shows the number of users' ratings by their accuracy level:

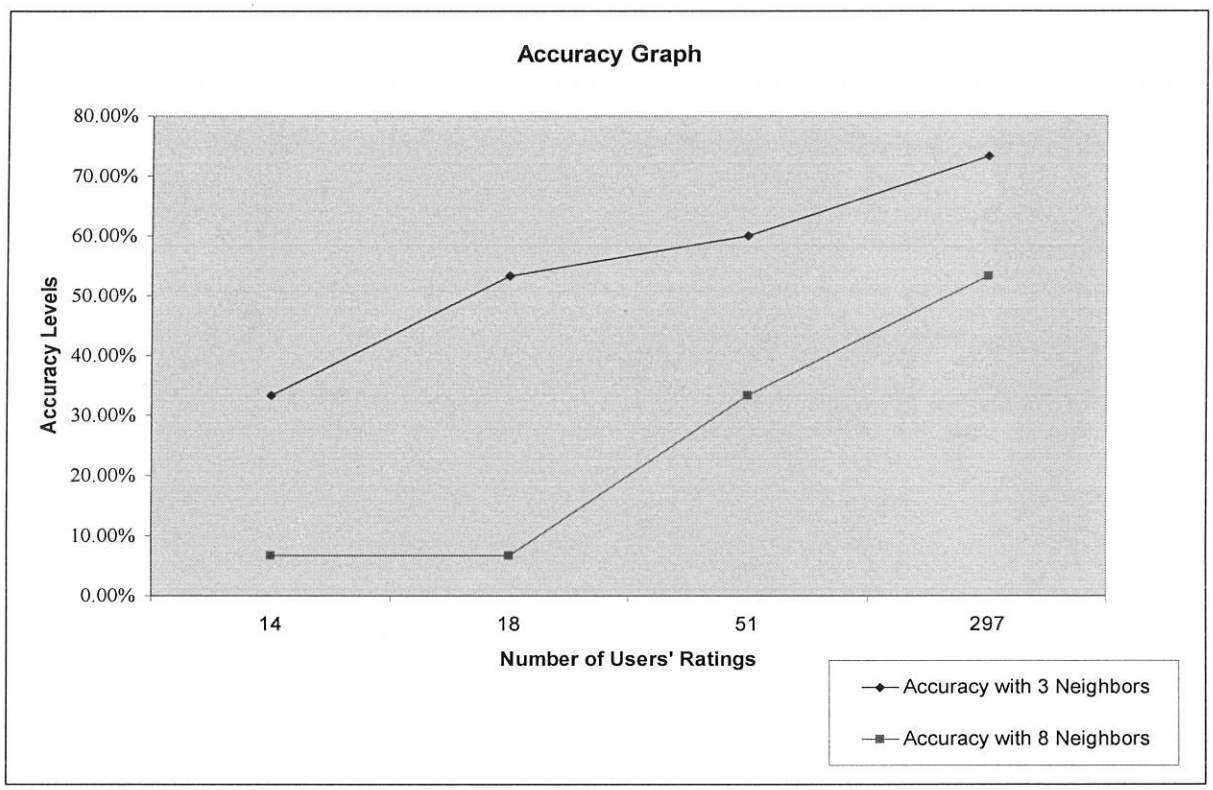

Figure 7: A Graph Showing Number of Users' Ratings by Accuracy Levels with 3 and 8 Best Neighbors

As we can see from the figure (figure 6) the accuracy for each user corresponds to the rating provided by that user. Users with low rating history are also the users with low accuracy. On the other hand, users who have good number of ratings are also those with high accuracy value.

This shows that in order to get a reasonably good recommendation, users have to provide as many ratings as possible. We can learn from this fact that collaborative filtering alone cannot be used in SDI systems since providing as many ratings as necessary requires too much users involvement. To be able to use collaborative filtering in SDI systems a way of implicitly learning the ratings without requiring much of users' effort have to be considered. To get better result it also needs to be integrated with content filtering agents and implemented in full software agent based approach where the agents learn the users' interest, filter information, and provide feedback and so on.

In general, the above results, although they are based on a very small number of test cases which can hardly be considered as representative of any sort, have showed that collaborative filtering is possible in SDI systems.

This study provided some indications on factors affecting the performance of a collaborative filtering system. These factors are mainly the history of rating of a user (the size of rating a user provided), the number of users and items in the database and the number of best neighbors selected.

Chapter Five

Conclusion and Recommendation

5.1. Conclusion

This thesis tried to explore problems related to SDI profile updating and has also attempted to demonstrate that collaborative filtering approach combined with the content filtering approach can be used to improve the situation. This research uses the ILRI SDI service data to verify whether the application of collaborative filtering would improve the efficiency of SDI services as it did in other information services.

Even though the developed system did not get a chance to be fully experimented and tested, there are indications that collaborative filtering can be a successful technique for SDI if combined with other agents.

The experiment was done based on the neighborhood algorithm. Pearson correlation coefficient was used as a similarity measure due to its wide spread acceptance throughout the literature. Number of best neighbors is also favored over correlation thresholding due to the indication in the literature that it is a better way of choosing neighborhood for the active user. Top N items are then recommended to a user by calculating the predictability of those items to the active user and ranking the items based on their prediction result.

The experiment was done using 32 users, 2352 items, and 2775 ratings. The output recommendations are sent to 10 users but feedbacks from only five users was received. The highest coverage of the system was 99% when all other users in the database (31 users) are

considered as best neighbors. The coverage becomes very low when the size of neighborhood drops to 8 and then drops further when only three neighbors are taken as best recommenders. However, the precision (accuracy) seems to be better, according to the result obtained from the five users, for neighborhood three rather than for eight.

Even though so many problems were faced during the process of the research, this study contributed to take the question of SDI service a step further. By starting from a practical problem the research explored in to the way of solving the problem using one technique.

5.2. *Recommendation*

This research paved the way to using collaborative filtering techniques for the purpose of improving the service provided by SDI systems. This study was not conducted to propose collaborative filtering alone as the omnipotent problem solving tool. Rather it starts by indication that the combination of collaborative filtering and content based filters is of paramount importance. Hence a lot of more researches should be conducted for the system (SDI service system) to be fully agent based. The full collaborative filtering functionality need to be also explored. For these reasons the following research areas are outlined for further research:

1. A more thorough experiment which considers all the important determinant factors (such as neighborhood sizes) that influence the performance of the system should be made.
2. The performance of the system should be tested with reasonable amount of user samples. A way of getting better number of feedbacks and using experts or users to judge all the relevant items for the purpose of calculation recall should be

brought to mind at the earlier stages of the research. This will ensure reliable generalization and conclusion to be made on the results.

3. The potentiality of the other similarity measuring techniques should also be discovered. That is, similarity measures other than Pearson correlation should be used to compare results and pick the best measure for the SDI case.
4. Model based algorithms such as clustering model and Bayesian network model must also be empirically tested in the pursuit for better performance.
5. The combination of collaborative filtering and content filtering has to be investigated into in the light of SDI systems. Researches have to be undertaken to combine the two systems for better result and efficiency of the SDI service.

REFERENCES

Abebe Chekol. 2001. *The Application of Interface Agent Technology for Personalized Information Filtering: the case of ILRIALERTS.* Addis Ababa University: SISA.

Altinel, M. M.J. Frankline. 2000. *"Efficient Filtering of XML Documents for Selective Dissemination of Information."* In Proceedings of VLDB Conference, Cairo. Available At: http://citeseer.nj.nec.com/altinel00efficient.html

Basu, Chumki, H. Hirsh, and W. Cohen. 1998. *"Recommendation as Classification: Using Social and Content-Based Information in Recommendation."* In Proceedings of the Fifteenth National Conference on Artificial Intelligence AAAI98. AAAI Press/MIT Press. Available at: http://citeseer.nj.nec.com/basu98recommendation.html

Billsus, Daniel and Michael J. Pazzani. 1998. *"Learning Collaborative Information Filters."* In Proceedings of ICML '98. pp. 46-53. Available at: http://www.citeseer.nj.nec.com/billsus98learning.html

Bose, H. 1986. *Information Science: Principle and Practice.* New York: Envoy Press.

Breese, J. S., D. Heckerman, and C. Kadie. 1998. *"Empirical Analysis of Predictive Algorithms for Collaborative Filtering."* In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43-52. Available at: http://citeseer.nj.nec.com/breese98empirical.html

Chau, M. et al. 2002. *"Design and Evaluation of a Multi-agent Collaborative Web Mining System."* In Decision Support Systems. Available at: http://citeseer.nj.nec.com/chau03design.html

Dawit Yimam. 1998. *Applying Interface Agent Technology to SDI user Profile Management: the case of ILRIALERTS.* Addis Ababa University: SISA.

Edmunds, A and A. Mooris. 2000. "*The Problem of Information Overload in Business Organizations.*" In International Journal of Information Management. 20(2000): 17-28.

Ferreira, J. and Alberto Rodrigues da Silva. 2001. "*MySDI: A Generic Architecture to Develop SDI Personalised Services (How to Deliver the Right Information to the Right User?).*" In Proceedings of the Third International Conference on Enterprise Information Systems (ICEIS'2001). Available at: http://berlin.inesc.pt/alb/papers/2001/iceis2001_jf_as.pdf

Geyer-Schulz, Andreas and Michael Hahsler. 2002. *"Evaluation of Recommender Algorithms for an Internet Information Broker Based on Simple Association Rules and on The Repeat-Buying Theory.*" In Proceedings WEBKDD. Available at: http://citeseer.nj.nec.com/geyer-schulz02evaluation.html

Goldberg, D. et al.1992. "*Using Collaborative Filtering to Weave an Information Tapestry.*" In Communications of the ACM. Available at: http://www.csl.sony.co.jp/person/masui/bib/Goldberg_Tapestry.html

Goldberg et al .2000. *Eigentaste: A Constant Time Collaborative Filtering Algorithm.* Available at: www.ieor.berkeley.edu/goldberg/pubs/eigentaste.pdf.

Good, N., et al. 1999. "*Combining Collaborative Filtering with Personal Agents for Better Recommendations.*" Proceedings of the 1999 Conference of the American Association of Artificial Intelligence (AAAI-99). (pp. 439-446). Available at: http://www.cs.umn.edu/Research/GroupLens/ papers/pdf/aaai-99.pdf

Griffith, Josephine and Colm O'Riordan. 2000. *Collaborative Filtering.* Available at: http://www.it.nuigalway.ie/cirg/publications.html

Herlocker, J.L. 2000. *Understanding and Improving Automated collaborative Filtering Systems.* University of Minnesota: Faculty of Graduate School. Available at: http://cs.oregonstate.edu/~herlock/papers.html

Herlocker,J.L., J.A. Konstan, and J. Riedl. 2000. "*Explaining Collaborative Filtering Recommendations.*" In Proceedings of the 2000 Conference on Computer Supported Cooperative Work. Available at: http://cs.oregonstate.edu/~herlock/papers.html

Herlocker, J.L., and et al. 1999. "*An Algorithmic Framework for Performing Collaborative Filtering.*" In Proceedings of the 1999 Conference on Research and Development in Information Retrieval. Available at: http://cs.oregonstate.edu/~herlock/papers.html

Kemp, A. 1979. *Current Awareness Service.* London: Bingley.

Konstan, J.A., et al., 1997. "*GroupLens: Applying Collaborative Filtering to Usenet News.*" In Communications of the ACM 40 (3), 77-87. Available at: http://cs.oregonstate.edu/~herlock/papers.html

Koubarakis, Manolis et al. 2001. *Efficient Agent-Based Dissemination of Textual Information.* Available at: www.intelligence.tuc.gr/~koutris/publications/setn02.pdf

Kumar, Krishan. 1978. *Reference Service. 2nd ed.* New Delhi: Vikas Publishing.

Kumar, Ravi. et al. 1998. *Recommendation Systems: a probabilistic analysis.* Available at: http://citeseer.nj.nec.com/kumar98recommendation.html

Lashkari, Yezdi. 1995. *Feature guided automated collaborative filtering.* Master's thesis, MIT Department of Media Arts and Sciences. Available at: http://citeseer.nj.nec.com/lashkari95featureguided.html

Lee, W. S. 2001. Collaborative Learning for recommender systems. In Proceedings of the Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers, pp. 314-- 321.Available at: http://citeseer.nj.nec.com/lee01collaborative.html

Leggate, P. 1975. "*Computer Based Current Awareness Services.*" In Journal of Documentation, 31(2): pp 93-115.

Lieberman, Henry.1997. *Autonomous Interface Agents.* Available at: http://lieber.www.media.mit.edu/AIA/AIA.html

Maltz, David and Kate Ehrlich.1995. *Pointing the Way: active collaborative filtering.* Available at: http://www.acm.org/sigchi/chi95/Electronic/documents/papers/ke_bdy.htm

Maltz, David. 1994. Distributed Information for Collaborative Filtering on Usenet NetNews. M.I.T. Department of EECS, Cambridge, Mass. Available at: http://www.b2b-machines.com/publications/sigma/multiagent.pdf

Maes,Pattie. 1997. "*Agents that Reduce Work and Information Overload.*" In Software Agents. Bradshaw, J. (Ed.), Cambridge, MA: MIT Press/AAAI Press.

Melville, Prem and Raymond J. Mooney and Ramadass Nagarajan.2002.*Content-Boosted Collaborative Filtering for Improved Recommendations*. In Proceedings of the Eighteenth National Conference on Artificial Intelligence(AAAI-2002),pp. 187-192, Edmonton, Canada. Available at: http://citeseer.nj.nec.com/melville02contentboosted.html

Miller,B.N., et al., 1999. *The GroupLens Protocol Specification*. Available at: http://www.cs.umn.edu/Research/GroupLens/protocol.html

Neill, S.D. 1992. *Dilemmas in the Study of Information: Exploring the Boundaries of Information Science.* New York: Greenwood Press.

Nwana H.S. 1996. *"Software agents: an Overview."* In The Knowledge Engineering Review, 11(3):1-40. Available at: http://www.cse.unsw.edu.au/~wobcke/COMP4416/readings/Nwana.96.pdf

Packer, K.H. and Dagobert Soergel. 1979. "*The Importance of SDI for Current Awareness in Fields with Sever Scatter of Information.*" In JASIS. 30(3):125-135.

Pao, M.L. 1989. *Concepts of Information Retrieval*. Colorado: Libraries Unlimited Inc.

Vel, O. and S. Nesbitt. 1998. "*A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web.*" In: Proceedings of Conference on Automated Learning and Discovery, Pittsburgh, PA, June 11-13, 1998, Carnegie Mellon University, Available at: www-dii.ing.unisi.it/~diligmic/Pub/esann.ps.gz

Wilson, Partike. 1995. "*Unused Relevant Information in Research and Development.*" In JASIS. 46(1995):45-51.

Whitehall, T. 1979. *Personal Current Awareness Service: a Handbook of Techniques for Manual SDI.* London: The British Library

Yan, T. W. and H. Garcia-Molina. 1999. "*The SIFT information dissemination system.*" In ACM Transactions on Database Systems, vol. 24, no. 4, pp. 529--565. Available at: http://citeseer.nj.nec.com/yan00sift.html

Yan. T. and Hector Garcia-Molina. 1997. "*Efficient Dissemination of Information on the Internet.*" In Data Engineering Bulletin 19 (3): 48-54. Available at: http://www.informatik.uni-trier.de/~ley/db/conf/pdis/YanG94.html

Yan T. and Hector Garcia-Molina. 1994. "*Distributed selective dissemination of information.*" In Proceedings of the Third International Conference on Parallel and Distributed Information Systems, pages 89--98. IEEE Computer Society. ftp://db.stanford.edu/pub/yan/1994/dsdi.ps

Yan, T. W. and H. Garcia-Molina. 1993. "*Index structures for information filtering under the vector space model.*" Technical Report. Stanford University, Stanford CA 94305. ftp://db.stanford.edu/pub/yan/1993/sdi-vector-model-tr.ps

Informants:

W/ro Maria Brunari

Ato Getachew Bultefa

Appendices

Appendix A: The Visual Basic Code

```
Option Explicit
Dim cnn As Connection
Dim rs As Recordset, rsUsers As Recordset, rsDesc As Recordset
Dim ActiveUID As Integer
Dim NbUsers As Integer

Private Sub cmdRecommend_Click()
   Dim cmd As New Command
   Dim ra As Recordset

 lstItems.Clear
 If vsNbNeighbor.Value > NbUsers Then
  MsgBox ("Invalid Best Neighbor Value!" & vbNewLine & vbNewLine & _
  "Try again!")
  cmdRecommend.Refresh
 Else
  lblStatus.Caption = "Processing..."
  lblStatus.Visible = True
  Refresh

  With cmd
     .ActiveConnection = cnn
     .CommandText = "qryUserAllItemsRatings"
     Set ra = .Execute(, ActiveUID)
  End With

  Dim meanRa As Single
  meanRa = mean(ra)

  ReDim ru(NbUsers) As Recordset
  ReDim meanRu(NbUsers) As Single
  ReDim w(NbUsers) As Single
  Dim i As Integer

  With rsUsers
     .MoveFirst
     For i = 1 To NbUsers
        Set ru(i) = cmd.Execute(, .Fields("User_ID"))
        meanRu(i) = mean(ru(i))

        w(i) = sum(ra, meanRa, ru(i), meanRu(i)) / (STD(ra, meanRa) * STD(ru(i), meanRu(i)))

        .MoveNext
```

```
    Next
End With

ReDim TopIndex(NbUsers) As Integer

Dim j As Integer, s As Single
Dim m As Integer, temp As Integer

For i = 1 To NbUsers
   TopIndex(i) = i
Next i

For i = 1 To NbUsers - 1
   m = i
   For j = i + 1 To NbUsers
      If w(TopIndex(j)) > w(TopIndex(m)) Then m = j
   Next j
   temp = TopIndex(i)
   TopIndex(i) = TopIndex(m)
   TopIndex(m) = temp
Next i

Dim nbTop As Integer

nbTop = vsNbNeighbor.Value

   Dim cmd2 As Command
   Dim rsitems As Recordset
   Dim aItems() As Long

   Set cmd2 = New Command
   For i = 1 To nbTop
      With cmd2
         .ActiveConnection = cnn
         .CommandText = "qryOthers"
         rsUsers.Move TopIndex(i) - 1, adBookmarkFirst
         Set rsitems = .Execute(, Array(ActiveUID, rsUsers("User_Id")))
         Add rsitems, aItems
      End With
   Next i
   Dim wSum As Single
   Dim NbOthers As Integer
   NbOthers = UBound(aItems)
   ReDim p(NbOthers) As Single

   For i = 1 To NbOthers
      For j = 1 To nbTop
```

```
            ru(TopIndex(j)).MoveFirst
            ru(TopIndex(j)).Find "Item_id = " & aItems(i)
            If IsNull(ru(TopIndex(j))("Rating")) Then
               s = s - meanRu(TopIndex(j)) * w(TopIndex(j))
            Else
               s = s + (ru(TopIndex(j))("Rating") - _
                  meanRu(TopIndex(j))) * w(TopIndex(j))
            End If
            wSum = wSum + w(TopIndex(j))
         Next j

         p(i) = meanRa + s / wSum

      Next i

      ReDim TopIndex(NbOthers) As Integer
      For i = 1 To NbOthers
         TopIndex(i) = i
      Next i

      For i = 1 To NbOthers
         m = i
         For j = i + 1 To NbOthers
            If p(TopIndex(j)) > p(TopIndex(m)) Then m = j
         Next j
         temp = TopIndex(i)
         TopIndex(i) = TopIndex(m)
         TopIndex(m) = temp
      Next i

    ' MsgBox UBound(aItems)
      lstItems.Clear
      i = 1
      While i <= UBound(aItems) And i <= vsNbItem
         lstItems.AddItem aItems(TopIndex(i))
         i = i + 1
      Wend
      lblStatus.Caption = "Done"
    End If
End Sub

Private Sub dtcUserID_Click(Area As Integer)
  If Area = 2 Then Update_Users
End Sub

Private Sub dtcUserID_Validate(Cancel As Boolean)
  If Trim(dtcUserID) = "" Then
    MsgBox "You must select a user first"
```

```
        Cancel = True
    ElseIf dtcUserID.Text <> ActiveUID Then
        rsUsers.Find "User_ID = " & dtcUserID.Text
        If rsUsers.EOF Then
            MsgBox "No user with this ID"
            Cancel = True
            dtcUserID_Click 2
        Else
            Update_Users
        End If
    Else
    End If
End Sub

Private Sub Form_Load()
  Set cnn = New Connection
    With cnn
        .CursorLocation = adUseClient
        .Open "Provider = Microsoft.Jet.OLEDB.4.0;" & _
              "Data Source = " & App.Path & "\ILRISDI.mdb;"
    End With

    Set rs = New Recordset
    Set rsUsers = New Recordset

    rs.Open "Ratings", cnn, adOpenDynamic, adLockOptimistic
    rsUsers.Open "Select Distinct User_ID From Ratings", cnn, adOpenDynamic,
adLockOptimistic

    Set dtcUserID.RowSource = rsUsers
    Set rsDesc = New Recordset
End Sub

Public Sub Update_Users()
    ActiveUID = Val(dtcUserID.Text)
    rsUsers.Close
    rsUsers.Open "Select Distinct User_ID From Ratings Where User_ID <> " & _
        ActiveUID, cnn, adOpenDynamic, adLockOptimistic
    Set dtcUserID.RowSource = rsUsers
    NbUsers = rsUsers.RecordCount
End Sub

Private Sub txNbItem_Validate(Cancel As Boolean)
        With txtNbNeighbor
        If Not IsNumeric(.Text) Or .Text < 1 Or .Text > NbUsers Then
            MsgBox "Invalid value"
            Cancel = True
```

```
        Else
            vsNbNeighbor.Value = .Text
        End If
    End With
End Sub

Private Sub lstItems_Click()
    On Error Resume Next
    With rsDesc
        .Close
        .Open "Select Desc From Items Where ItemId = " & lstItems.Text _
            , cnn, adOpenDynamic, adLockOptimistic
        Set txtDesc.DataSource = rsDesc
    End With
End Sub

Private Sub txtNbItem_Validate(Cancel As Boolean)
    With txtNbItem
        If Not IsNumeric(.Text) Or .Text < 1 Or .Text > 10 Then
            MsgBox "Invalid value"
            Cancel = True
        Else
            vsNbItem.Value = .Text
        End If
    End With
End Sub

Private Sub txtNbNeighbor_Validate(Cancel As Boolean)
    With txtNbNeighbor
        If Not IsNumeric(.Text) Or .Text < 1 Or .Text > NbUsers Then
            MsgBox "Invalid value"
            Cancel = True
        Else
            vsNbNeighbor.Value = .Text
        End If
    End With
End Sub

Private Sub vsNbItem_Change()
    txtNbItem = vsNbItem.Value
End Sub

Private Sub vsNbNeighbor_Change()
    txtNbNeighbor = vsNbNeighbor.Value
End Sub

'Functions/Modules
```

```
Public Function mean(rs As Recordset) As Single
   Dim s As Single
   Dim count As Integer
   s = 0
   count = 0
   With rs
      .MoveFirst
      While Not .EOF
         If Not IsNull(.Fields("Rating")) Then
            s = s + .Fields("Rating")
            count = count + 1
         End If
         .MoveNext
      Wend
      mean = s / count
   End With
End Function

Public Function sum(rs1 As Recordset, mean1 As Single, _
   rs2 As Recordset, mean2 As Single) As Single

   Dim val1 As Integer, val2 As Integer
   With rs1
      .MoveFirst
      rs2.MoveFirst
      While Not .EOF
         If IsNull(rs1("Rating")) Then
            val1 = 0
         Else
            val1 = rs1("Rating")
         End If

         If IsNull(rs2("Rating")) Then
            val2 = 0
         Else
            val2 = rs2("Rating")
         End If
         sum = sum + (val1 - mean1) * (val2 - mean2)
         .MoveNext
         rs2.MoveNext
      Wend
   End With
End Function

Public Function STD(rs As Recordset, mean As Single) As Single
   Dim s As Single
```

```
    s = 0
    With rs
      .MoveFirst
      While Not .EOF
        If IsNull(.Fields("Rating")) Then
          s = s + mean * mean
        Else
          s = s + (.Fields("Rating") - mean) * (.Fields("Rating") - mean)
        End If
        .MoveNext
      Wend
    End With
    STD = Sqr(s)
End Function

Public Sub Add(rs As Recordset, a() As Long)
    Dim i As Integer, n As Integer
    Dim found As Boolean

    On Error Resume Next
    n = UBound(a)
    With rs
      .MoveFirst
      While Not .EOF
        i = 1
        found = False
        While (i <= n And Not found)
          If rs("Item_Id") = a(i) Then
            found = True
          Else
            i = i + 1
          End If
        Wend
        If Not found Then
          ReDim Preserve a(n + 1)
          a(n + 1) = rs("Item_Id")
          n = n + 1
        End If
        .MoveNext
      Wend
    End With

End Sub
```

Appendix B: The Access Queries

qryItems

```
(SELECT DISTINCT Item_Id
FROM Ratings;)
```

qryOthers

```
SELECT  Item_Id, Rating
FROM Ratings
WHERE User_Id=User2 And
        item_id  not in (select item_id
                         from ratings
                         where User_Id =User1;)
```

qryUserAllItemsRatings

```
SELECT qryItems.Item_Id, Rating
FROM qryItems LEFT JOIN qryUserRatings ON
qryItems.Item_Id=qryUserRatings.Item_Id;
```

qryUserRatings

```
SELECT Item_Id, Rating
FROM Ratings
WHERE User_Id=User;
```

Appendix C: Sample output sent to user_id 010

Article ID	Article Descriptions	Relevance (Yes, No, Can't Say)
64021	"Sevinc, F.; Kamburgil, K.; Dik, B.; Guclu, F.; Aytekin, H." "Konya yoresinde atik yapan ve yapmayan koyunlarda Indirekt Fluoresan Antikor (IFA) testi ile toxoplasmosis arastirmasi" "[The seroprevalence of toxoplasmosis by indirect fluorescent antibody (IFA) test in ewes with and without abortion in Konya province]" "Saglik Bilimleri Dergisi, Firat Universitesi" "v. 14(1)" "p. 137-142" "35 ref." "2000" "KENYA; SHEEP; TOXOPLASMA GONDII; ABORTION; AGE; ANTIBODIES; EWES; SEROPREVALENCE" "This study was carried out to determine the seroprevalence of toxoplasmosis by IFA test in healthy ewes and ewes with abortion in Konya province. The blood samples were collected from 283 ewes were abortion and from 827 healthy ewes in Cumra, Kadinhani and Altinekin districts between October 1996 and December 1997. The serum samples were examined for the presence of anti-Toxoplasma gondii antibodies by IFA test. Anti-Toxoplasma gondii antibodies were detected at titres of >= 1/64 in 13.78% and 10.16% of the sheep with and without abortion , respectively. These results showed that there was no significant difference between two groups ($p > 0.05$). In the present study, the antibody titres were varied from 1/64 to 1/2048. There was a correlation between the seroprevalence of Toxoplasmosis and the age of the sheep."	
64084	"Dinesh Patel; Misraulia, K.S.; Reddy, A.G.; Garg, U.K.; Sharma, R.K.; Bagherwal, R.K.; Gupta, B.K." "Effectiveness of artemether in induced bovine tropical theilerosis in crossbred calves" "Indian Veterinary Journal" "v. 78(5)" "p. 386-389" "13 ref." "2001" "CATTLE; HYALOMMA ANATOLICUM; THEILERIA ANNULATA; BOVINE TROPICAL THEILEROSIS; ARTEMETHER; CALVES; CROSSBREDS" "This study evaluates the	

Declaration

This thesis is my original work and has not been submitted as a partial requirement for a in any other university.

Zehara Zinab

July, 2003

This thesis has been submitted for examination with our approval as university advisors

Tesfaye Biru

Workshet Lamnew

Ethiopia Taddese