JIMMA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF INFORMATION SCIENCE

DESIGNING A SELECTIVE DISSEMINATION OF INFORMATION (SDI) SERVICE FOR ACADEMIC LIBRARY: THE CASE OF JIMMA UNIVERSITY LIBRARY SYSTEM

BY

MISGANU FEKADU

OCTOBER, 2016

JIMMA, ETHIOPIA

JIMMA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF INFORMATION SCIENCE

DESIGNING A SELECTIVE DISSEMINATION OF INFORMATION (SDI) SERVICE FOR ACADEMIC LIBRARY: THE CASE OF JIMMA UNIVERSITY LIBRARY SYSTEM

A Thesis Paper Submitted to Department of Information Science, College of Natural Science, Jimma University, in Partial Fulfillment for Award of MSc. Degree in Information Science (Information and Knowledge Management)

By

Misganu Fekadu

Advisors:

Principal Advisor:-Million Meshesha (PhD)

Co-Advisor:- Samuel Sisay (MSc)

October, 2016

Jimma, Ethiopia

JIMMA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF GRADUATE STUDIES

DEPARTMENT OF INFORMATION SCIENCE

DESIGNING A SELECTIVE DISSEMINATION OF INFORMATION (SDI) SERVICE FOR ACADEMIC LIBRARY: THE CASE OF JIMMA UNIVERSITY LIBRARY SYSTEM

BY:

MISGANU FEKADU

As members of the board of examining of the MSc thesis open defense examination of the above title held on October 22, 2016, we members of the board those our name are listed below, were read and evaluated the thesis and examined the candidate.

Name	**Title**	**Signature**	**Date**
	Chairperson		
Million Meshesha (PhD)	Principal Advisor		
Samuel Sisay (MSc)	Co-Advisor		
	External Examiner		
	Internal Examiner		

DECLARATION

I declare that this thesis is my original work and it has not been presented for any degree in any other universities. All the material sources used in this work are properly appreciated.

_____

Misganu Fekadu

October, 2016

This thesis has been submitted to the department for examination with our approval as university advisors:

Principal Advisor: Million Meshesha (PhD) _____

Co-Advisor: Samuel Sisay (MSc) _____

October, 2016

DEDICATION

I dedicated this work to my beloved wife Aster Debela

ACKNOWLEDGMENT

Above of all, I would like to thank almighty God for giving me strength through my whole life, for opening my mind to propose this work and for helping me to complete the work. Next, my gratitude is forwarded to my advisor; Dr. Million Meshesha for his continues help from idea generation of selecting topic to the end of this thesis work and for his valuable comments and solutions during difficulties.

I also express my acknowledgment to my Co-advisor Ato Samuel Sisay for his giving me direction through out of this work. His comments were constructive and have been paramount contribution for the completion of this thesis.

I also want to thank my wife, Aster Debela for her making me relax when I become weak to continue the work with new mind and for her whole inexpressible support and encouragements.

I would not like to pass without giving my thanks to Jimma University for giving me the opportunity of learning this MSc program and for providing enough broadband internets without which this research would be incomplete.

Finally, I would like to present my gratitude to JULS staff and my classmates since their contribution is not less for the completion of this work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CAI	**Current Awareness Information**
CBCF	**Content-Boosted Collaborative Filtering**
CBF	**Content-Based Filtering**
CF	**Collaborative Filtering**
CSV	**Comma separated values**
DF	**Document Frequency**
FSM	**Finite State Machine**
IDLE	**Interactive Development Language Environment**
IF	**Information Filtering**
JiT	**Jimma University Institute of Technology**
JULS	**Jimma University Library System**
LSA	**Latent Semantic Analysis**
LSI	**Latent Semantic Indexing**
MAP	**Mean Average Precision**
NLTK	**Natural Language Toolkit**
OPAC	**Online public access Cataloging**
SDI	**Selective Dissemination of Information**
TF*IDF	**Term Frequency*Inverse Document Frequency**
UDL	**University Digital Library**
VSM	**Vector Space Model**
XML	**eXtended Markup Language**

ABSTRACT

Library acquires resources from time to time to have a balanced resource within the increment of library users and to have the newest resources as soon as they are published especially in academic libraries. As resources increase, it becomes difficult to users to select important references and information of their interest. Therefore, this study aims to design Selective Dissemination of Information (SDI) service that provides information alerting service to keep individuals informed of new resources (books, article, etc) in their particular fields of interest. Design science research method (DSRM) which creates and evaluates IT works proposed to solve recognized organizational problem and the process of inspiring, designing, demonstrating, evaluating, and communicating the artifact was followed. To this end, a prototype SDI system is developed to recommend arrival of new books and journals using python programming language for Jimma University Library System (JULS) users by applying an information filtering approach. Concerned population for this study was 1610 academic staff of Jimma University; out of which 921 considered for sample selection since the rest users have no staff profile on Jimma University website. Hence, Profiles of eighty-six (9.34%) academic staff were registered in user database. Among these eighty-six users, twelfth (12%) of them were used for user acceptance testing. MySQL version 5.5 was used for recording user profile. For testing the prototype SDI system, twenty percent of the data is used and the rest data is used for training. In this work, different matching schemes are experimented; among them, TF*IDF weighting technique with Vector Space Model (VSM) has registered the best performance of 78.76% precision. In addition, the SDI system achieves 95% in user's acceptance testing which shows that it has high user's acceptance. This further means that, it is advisable to use SDI system to enhance library services. However, the proposed prototype system does not recommend books written in other than English language. Therefore, future research direction is to develop an SDI system that recommends books written in different languages, including local languages.

Keywords: Academic Library, Selective Dissemination of Information, Vector Space Model

CHAPTER ONE

INTRODUCTION

1.1 Background

Due to the increment of users and new inventions as well as technological enhancement, it is mandatory for libraries to have many collections, including books, journals and other resources. Especially academic libraries in the higher education are expected to enhance their collection and provide user-friendly services.

As a result, the library collections become larger from time to time especially in academic library and users encounter difficulties in obtaining specific information with their interest. Librarians have become increasingly concerned with how best to provide their clientele with adequate means of keeping current with the literature of their subject area (Wood & Seeds, 1974). Because of the volume of literature published in different fields, it is especially, important for academic librarians to be able to provide an adequate form of service, which can help users, find information with their areas of fields. One of the major functions of a library is to analyze publications immediately after receipt, make a selection of publications pertinent to the program of the organization and bring individual item to the attention of the user to whom it may concerns (Rao, 1993).

This is why libraries have been a focal point for implementing Selective Dissemination of Information service, which has been used to provide users with updates of bibliographic information (O'Neil, 2001).

Selective dissemination of information (SDI) solves the problem of obtaining specific information with one's interest. In SDI scenario, information producers publish information to an SDI service and this information is forwarded to information consumers that have already subscribed to it with a matching profile (Koubarakis & Koutris, ND.).

As Rao (1993) described, SDI is information notifying service designed to keep individuals informed of new developments in their particular fields of interest by providing a listing of

citations to newly acquired literature, based on predefined statements of interests commonly known as profiles.

As Hossain & Islam (2008) said, SDI is a current awareness mechanism through which a user can expect to receive notification of new items and data in accordance with his/her statement of interest or profile.

The concept of Selective dissemination of information (SDI) is an old one in which librarians have been providing SDI service on manual basis for long time (Ababor, 2003). However, as the number of documents hold by information provider (Library) becoming large and user's interest are changing from time to time, librarians are faced difficulties in providing SDI service manually. It is at that time (1950's) when researchers are formulated idea of computer based SDI service that make it likely to offer users potentially important documents by accessing users' personal information need kept by the library (Ababor, 2003; Morales-del-Castillo et al., 2013).

The earlier concept of computerized 'SDI' by Luhn (1958) now has undergone a radical change due to the latest revolution of computer and telecommunication technologies joined with the present library services (Hossain & Islam, 2008; Ababor, 2003).

The aim of an SDI system is to bring new information arriving at information provider, library to users who express their interests via user profiles (O'Neil, 2001). As Ababor (2003) noted, the main objectives of SDI service is to help 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. SDI is primarily concerned with keeping users informed with information matching a user profile as it arrives at an information provider (O'Neil, 2001).

An efficient SDI is expected to provide as few non-relevant documents and as much relevant documents as possible. Relevant information or documents are identified and classified based on the long lasting interest of users which is called interest profile (Ababor, 2003).

Therefore, the problem of SDI is related with the issues of other systems like information filtering (IF) and information retrieval (IR). These systems are basically targeted in selection of a subset of documents from the document set, which are relevant to a user interest (Ababor, 2003). Thus, the main concern of this research is providing SDI service with information filtering

approach since Information filtering (IF) systems are designed for long term users with long term information need and for repetitive usage (Renganathan, Ajit & Suptendra, 2013; Hanani, Shapira, & Shoval, 2001). However, this difference of IF from IR is only at user's side. That means, IF and IR is more similar at author (developer) side. IF can be based on all IR models like Vector space Model (VSM) (Aberer & De, 2006; Shuda, Jiangping, & Riu, 2009); specially for matching of information needs with information items which is the heart of information retrieval model. In this scenario, IR approach is needed in the development of SDI service and hence we cannot separate IR from IF.

Authors, such as Belkin & Croft, (1992) and Porcel, Moreno & Herrera-Viedma, (2009) argued that SDI is similar with information filtering in that it keeps information to selectively flow to the interested user instead of making the user to go after the information.

Information filtering deals with the delivery of information that the user is likely to find interesting or useful. An information filtering system helps users by filtering the data source and deliver relevant information to the users. When the filtered information comes in the form of suggestions, the system is called a recommender system. Because users have variety of information need, the information filtering system must be modified to accommodate specific user's interests. This requires the gathering of feedback from the user in order to make a user profile of his preferences.

The process of information filtering needs maintaining user profiles, which are representations of that user's interest/information needs. From user's Profile, these information need/interests are descriptors or keywords that was matched with indexed document (document descriptor) a user needs to retrieve. Filtering is based on statements of individual or group information need, often called profiles (Belkin & Croft, 1992).

Based on the data acquisition, there are three major approaches for information filtering: Content-based filtering, collaborative filtering and knowledge-based information filtering (Spiegel, 2009). A content-based filtering system selects items based on the intersection between the content of the items and the user's information need, while a collaborative filtering system chooses items based on the intersection of items rating between people. A knowledge-based information filtering is one that uses knowledge about users and products to follow a knowledge-

based approach to making a recommendation, reasoning about what items meet the user's information needs (Burke, 1999). It is also possible to combine any two of these three approaches to develop a Hybrid system (Zhang, Min, He, & Xu, 2015; Spiegel, 2009; Wang, Xie, & Li, 2007).

1.2. Jimma University Library System

Jimma University library system (JULS) categorized under academic library of type of library since academic library is one type of library that is attached to a higher education institution, which serves two complementary purposes to support the school's curriculum, and to support the research of the university faculty and students.

Above all, academic library should be facilitated with ICT tools and infrastructure to provide information since the main purpose of an academic library is education and research (Rah, Gul, & Wani, 2010).

The primary objective of Jimma University Library system is to provide well-organized information resources, services & facilities for academic, research and other purposes to users in support of the program of the university.

The JULS has the following eight branch libraries[1]:

- Social Sciences Library
- Agriculture and Veterinary Medicine College Library
- Health Sciences Library
- Natural Sciences Library
- Technology Library
- Graduate Studies Library
- Law Library and
- Female Students Library

[1] https://www.ju.edu.et/library/

As known, Jimma University library system serves whole community of the University including staff and students. Therefore, JULS has about forty-nine thousand and two hundred total users from which 7200 were staff and 42000 were students.

JULS houses around 200,000 books, physical journals, magazines and audiovisual collections in all branches. It has also access to7200 online e-journal resources (full-text documents, reviews, abstracts, and databases), in addition to over 10 million digital off-line eGranary digital library.

Thus, to achieve its objective it is important for JULS to provide system that supports users in finding current information within their specific interest.

1.3 Statement of the problem

Jimma university Library system is the one among leading University libraries in the country. It acquires books in different ways; most of the time through purchasing and donation. On average, the library acquires at least Five thousand materials annually (JULS, 2014). With the existing system, it is difficult to inform users to know these materials as soon as they arrived at library system for use. Though making users aware of newly arrived documents are not emphasized well, JULS uses different mechanisms (JULS, 2014).

Jimma University Library System (JULS) uses manual current awareness service, called "new arrival display" that attaches bibliographic lists of newly arrived books on the notice board. However, this manual system did not solve user's specific information need due to two reasons: First, lists of bibliographic information are not specific to any user since the display is all lists of information arrived at library and second, it is not convenient for users to go to notice board and see the information since it consumes user's time.

Sometimes JULS also sends lists of delivered books through email for each college especially when purchase is carried on. Beyond adding burden to college to announce new arrivals for departments, this mechanism still did not solve the problem of obtaining current and suitable/user specific information about books from the root specifically, with their interest for users since mailed list is general one. This is almost similar with the existing "new arrival display" except it goes at the windows of users.

Jimma University Library System also uses library technologies like library automation. By using this automation, users are able to get information they need especially through online public access catalog (OPAC). In addition to making of users to go after the information, still there is a gap with this service in aiding users finding their information need since user effort is required to find information. That means, OPAC did not give information for new development rather than retrieving information from whole document without time interval by typing queries to find information.

Due to their deficiencies, having those systems mentioned above, it is difficult to reach users and they are becoming far from the library service. Most of the time, users specifically, the academic staff are complaining about lack of resources at library. This is not because of unavailability of resources, but they have no information about what resources at the library. Lack of information when books are received at library made them believe that there are no resources. This further shows that users are far from library. Even if, new edition of books exist, users are still busy with elder editions.

It is thus significant to build SDI service that helps JULS manage and maintain user interest profiles that better describes their information need and brings recommendation of new books accordingly. This allows the Library systems to serve the information needs of its users better.

There are researches done for SDI system by using information filtering (recommender system) for different organizations. Scholars like Spiegel (2009), used online movieLens dataset for their work, hybrid recommendation system (IF) for movie rental. Most scholars did SDI system to recommend information from web pages (O'Neil, 2001; Wang, Xie, & Li, 2007; Yun, Xun, & Huamao, 2008), while some scholars, Morales-del-Castillo et al. (2013) and Porcel & Herrera-Viedma, (2010) propose SDI system for Digital library. Amazon recommender system is an example of e-comerce recommendation system wich uses item-to-item collaborative filtering and recommends a user an item that is related to item which user is selected after retrieval (Linden, Smith, & York, 2003)

These researches are done by using content-based and collaborative recommender system. However, these two approaches have their own drawbacks. For example, in content-based information filtering, users are unable to update their information. That means, system could

only discover the information similar to user's current interests, efficiency and quality is much reduced in a long term information need and in collaborative filtering, user information item matrix is sparse if information items exceeds what users absorb and system performance is lower with increment of users and information sources (Wang et al., 2006).

To overcome limitations of those approaches (content-based and collaborative information filtering), different scholars (Wang et al., 2006; Spiegel, 2009; Porcel, Tejeda-Lorente, Martinez, & Herrera-Viedma, 2012; Porcel et al., 2012) combine these two approaches and proposed hybrid information filtering (recommender system). Hybrid recommenders are systems that integrate multiple recommendation techniques together to achieve a synergy between them (Spiegel, 2009).

However, all information filtering particularly recommender systems mentioned above is based on explicit data (user's rating of items) or implicit data (click or browse history). That means there should be large amounts of products or items provided for users to rate or browse and users are abundantly engaged in providing input for recommender system based on item provided because, system with a small base of ratings is unlikely to be very useful (Burke, 1999).

Knowledge-based information filtering (recommender system) is another type of recommender system which does not depend on user ratings and applied in the circumstance that is difficult to apply other information filtering approaches (Burke, 1999). This approach uses knowledge about users and products to follow a knowledge-based technique to generate a recommendation, reasoning about what products meet the user's information need. In this approach, even though users are not engaged in rating of items, still there is a need to involve users to answers some questions about product from resource catalog to capture exact knowledge of users or to have detail information about product what users are searching for. Sometimes users provide queries (detail information about product) and the system recommends a product similar to query based on detail information obtained from user about the product. Most of the time, Knowledge-based information filtering techniques support product consumers and sales representatives in the identification of appropriate products and services (Felfernig, Isak, Szabo, & Zachar, 2007)

So, the intension of this study is to design SDI service for academic library without any involvement of users about items at library because in SDI scenario, no need to have

specification about items since the recommendation is new items. Hence, the aim of this research is to develop SDI service that recommends newly arrived books based on knowledge-based information filtering approach for Jimma University library system.

To this end, this study tried to explore and answer the following research questions.

- What are the basic attributes of documents and user profiles that can help to recommend bibliographic lists of books?
- Which information-filtering model is best to recommend bibliographic lists of books?
- To what extent the proposed system makes targeted recommendation of newly arrived books for JULS users?

1.4 Objective of the study

1.4.1 General objective

The main objective of this study is to design SDI service that makes recommendation of newly arrived books for Jimma University Library System (JULS's) users by applying an information filtering approach

1.4.2 Specific objectives

To achieve the general objective of this study, the following specific objectives are formulated.

- To formulate user profile based on user information needs
- To identify basic attribute from document and user profile that can properly represent documents and users for recommending bibliographic lists of books.
- To identify information filtering models which can best recommend bibliographic lists of books
- To develop a prototype SDI system that recommends newly arrived books to users based on their profile
- To evaluate the performance and user acceptance of proposed prototype SDI system

1.5 Scope and limitation of the study

Currently, JULS is running deferent projects such as Digital Library and Institutional Repository. The scope of this study focuses on providing a prototype SDI system using knowledge-based information filtering to make recommendation of bibliographic lists of newly arrived books.

Even though knowledge-based IF is based on knowledge of users about product (data acquired and given to the system from users about product), this system is different from existing knowledge-based IF in data acquisition (no need of users to give information about the product to SDI system). However, the SDI system can be developed in an assumption that library knows user's interest and provides products based on user's profile.

The prototype SDI system proposed in this study is able to recommend new arrived books to registered users according to their interest. However, the system does not recommend books published in local languages like Afan Oromo and Amharic. As it is known, the instruction medium is English in higher education in Ethiopia and as a result, most of books are collected in English than other languages. Therefore, this is the first reason why this study attempted to provide SDI system in English. That means, the formation of such system is dependent on the language in which documents are written. So, linguistic processing takes place in the creation of SDI system.

The second reason is that during data acquisition specially, on the Jimma University website, there is no user set his/her user profile in any language except English.

1.7 Significance of the study

Information is a very vital tool for maintaining a healthy society and sustaining stable development in all surfaces of life (Nkiko & Iroaganachi, 2015). If information is very vital**,** it is not difficult for anyone to think the values of latest information**;** it is excellent instrument to maintain healthy society. SDI service is beyond giving information but it is all about giving latest information.

SDI service alleviates the burden of searching for information and brings current information at user's window based on their predefined information need. It also reduces burden of librarian in giving information.

SDI service needs to channel the huge accessible information as per clients' interest. This will empower JULS to satisfy goal of keeping clients informed of the newly development in their separate fields of interest. It will likewise permit the clients find what they need in a considerable arrangement of information.

Therefore, the SDI system that was developed is expected to benefit the following bodies:

- Library Community especially academicians by aiding them in finding relevant information timely.
- Library itself since the concern of library is how best to provide their clientele with adequate means of keeping current with the literature of their subject area.
- It can also serve as an input for researchers who want to study in this area.

1.8 Methodology of the study

Methodology refers to the principles, procedures, and practices that govern research and encompassing the entire process of conducting research (Marczyk, DeMatteo, & Festinger, 2005). Therefore, to achieve the main objective of this study the following step by steps procedures are followed.

1.8.1 Research Design

In this study, design science research was followed. Design science creates and evaluates IT artifacts intended to solve identified organizational problem and the process of inspiring, designing, demonstrating, evaluating, and communicating the artifact is consistent with the Design Science Research Method (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). As per the design science research, this study passes through Problem identification, define the objectives for a solution, design and development (prototype SDI service in this case), demonstration, evaluation and communication (writing report).

1.8.2 Study Area

This study was conducted for Jimma University Library System (JULS). JULS is established in line with establishment of the University in 1999 with nomenclature of Jimma University library system as main branch. JULS supports other branches such as Agriculture and Veterinary Medicine College Branch library, Health Sciences Branch library, Education Branch library, Technology Branch library, Social Sciences Branch library, Graduate Studies Branch library and Law Branch library by human resource, acquisition of resources (printed and electronic), other infrastructures and also automation of library and Digital library/Institutional repository is carried out under JULS rather than at branch library.

1.8.3 Source of data

To have deep understanding in the area of selective dissemination of information and identify the gap that is not covered by previous studies, different materials; including research works, journal articles, eBooks, and the Internet are reviewed.

To obtain user interest and construct user's profile, data was collected from secondary source. In this study, library user's interest was gathered from Jimma University website, http://www.ju.edu.et. The researcher tried to collect information directly from users by providing form on which users fill their profile. However, there is resistance with users to fill their personal information such as name and Employee Id. Therefore, since the researcher phase this problem, it is mandatory to find secondary data that has full information about users or user profile. That is why Jimma University website was used as a source of data. Using secondary data is also advantageous in saving time than collecting information from each user. Because no losses of selected user from website but during direct collection of data from users, that selected user may not be at office unless appointment is taken.

1.8.4 Data set

A data size of around 500 KB (0.5 MB) Bibliographic lists of books received by library acquisition from purchased and donated materials was used as data set for this study. Bibliographic information arrived at library acquisition are registered using Microsoft Excel and covers all domains. That means, lists of documents belongs to all departments was covered.

However, for this study the file format was changed from EXCEL to CSV (Comma separated values).

1.8.5 Study population and sampling method

Academic staffs were considered as whole population for this study. This is due to the fact that information need is expected dynamic with academic staff. To know the total number of academic staff first we collected lists of them from each colleges and institute. After we sum up all lists from each college, we found one thousand six hundred ten (1610) of academic staffs.

To train the SDI system, the researcher selects high ranked academic staffs. When we say high ranked, it is relative comparison of staffs, may be by their educational level or status. Let's say, in one department if there is no PhD holder, and if there is one staff promoted to assistant professor, that assistant professor is considered as high ranked staff in that department. The comparison is done from first degrees to PhD level as well as from assistance Professor to professor. That means if staffs have the same education level and different academic status, the staff that has more status has been selected. Because, from the researcher observation, high ranked academic staffs have putted their research interest properly than low ranked staff.

After high ranked users are identified, two of users (academic staffs) were selected purposely from each department whose have staff profile to provide user profile of them. Staff profile for some departments is under construction and therefore, not considered for this work. For example, college of education and behavioral science, some departments of JiT and Public health and Medical Science have no staff profile on university's website. Total numbers of staffs that have no staff profiles on the Jimma university website are six hundred eight-nine (689). So we subtracted these staffs from our population and therefore, nine hundred twenty one (921) academic staffs were considered for sample selection. In this study, no need of thinking whether the user represents whole users or not since document is recommended individually to users based on their interest. This is why Purposive sampling is selected. To train the system, profile of eighty-six (86) users were registered in the user database which is 9.34% of total population. Among registered users ten (12%) of them are selected for final user testing of the proposed system.

Table 1.1 Summary of samples taken for training and testing by college/institute.

College/Institute	Population	Samples taken	Samples in %
Business and Economics College	77	8	10.4
College of Agriculture and Veterinary Medicine	132	12	9.1
College of law and Governance	53	4	7.5
College of Natural Science	135	14	10.4
Social Science College	127	14	11
College of Health and Medical Science	234	22	16.4
Jimma institute of Technology	163	12	7.4
Total	921	86	9.34

1.8.6 Development Tools

For prototype SDI system, the development platform and programming language used are Windows environment and Python 2.7.11 respectively. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be extremely readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical dependencies than other languages (Tutorials Point, 2014). Python is also suitable for handling text operation and boasts a fast, high quality library for similarity computing, gensim (Rehurek & Sojka, 2010). Python is so far powerful programming language with excellent functionality for processing linguistic data (Bird, Klein & Loper, 2009).

For this research work, a database is constructed using MySQL database version 5.5 for recording user's profiles. Because, MySQL is an open-source database management system with a feature of easily importing file from external sources such as CSV file format. It is also the most popular database tools; easy to find support online.

1.8.7 Testing techniques

The evaluation processes focus on the system performance and user acceptance testing.

System performance testing techniques includes Precision, Recall and F-measure.

Precision is the fraction of recommended items that is actually relevant to the user while Recall is the fraction of relevant items to the set of recommended items. F-measure helps to simplify precision and recall into a single metric (Isinkaye, Folajimi, & Ojokoh, 2015).

To test the ability of different matching models to recommend documents, researcher used top ten ranked and recommended documents. Because SDI is evaluated by its ability to filter out related documents on the top so that users got relevant documents easily (Hossain & Islam, 2008).

User's acceptance testing is on the other hand enables to understand users feeling about the system. By providing questionnaire for users the researcher make sure whether they would like to use the proposed system or not, and to what extent the proposed system will meet users specific information need.

1.9 Operational definitions of terms

In this study, some terms were used interchangeably such as information need and subject interest, documents, items and new arrival books

New arrival - new document/book coming to the library

User's information need/Subject interest - statement describes subject area of users. It is used to determine what document the user is looking for.

User profile: is a database containing records of users including user's information need

1.10 Organization of the document

This study encompasses six chapters. Chapter one discusses background of the study, the problem statement, general and the specific objectives of the study, and methodologies that the researcher used to conduct this study

Chapter two is more about literature review both conceptual and related works. Under conceptual review, concepts related to SDI and IF is discussed abundantly including purposes, methods, technologies and procedures used.

Chapter three discusses about conceptual modeling of SDI, data and methods. Things under data are about preparation of documents and user profile.

Chapter four is implementation. This chapter describes what is implemented and how each task is implemented one by one specially in programming language called python.

Chapter five is experimentation and Evaluation result. Each matching models implemented under chapter four were experimented and evaluated both by system and users. Under this chapter, discussion was also made on results found that shows strength and weakness of the system. The last chapter is conclusion and recommendation.

CHAPTER TWO

LITERATURE REVIEW

2.1. Overview of SDI Service

SDI from Library Science is a current awareness system that alerts individuals to the latest publications in their specified field(s) of interest[2]. Starting from Luhn (1958), SDI has been defined by different researchers in almost similar ways:

According to Luhn, SDI is a service within an organization, which concerns itself with the channeling of new source of information, from whatever sources, to those clients within the organization where the probability of usefulness in connection with latest work or interest is high.

Bivona & Goldblum (1967) give operational definition of SDI by incorporating auto-indexing or controlled indexing vocabulary (index terms are selected directly from the texts of abstracts or titles based on frequency of occurrence, word length, gross grammatical structure) which is missed from Luhn definition of SDI. Therefore, Bivona and Goldblumas defined SDI as a system, which has:

- **Input** to the system consists of any information (for dissemination) which can be characterized by a string of characters and which can be graphically reproduced, descriptions of users' information requirements, a list of the users' addresses and feedback from the users which indicates the degree of relevancy of output received and provides a basis for improving relevancy of output.
- **Processes performed by the system** consist of matching descriptions of users' information requirements against descriptions of the contents of input documents, selection of document descriptions, which match users' information requirements, modification of users' profiles based on feedback from the users and addressing of outputs selected.

[2]http://dictionary.reference.com/browse/selective-dissemination-of-information

- **Output** consists of document descriptions selected and addressed in process and statistics, which indicate the operational features of the system.

"*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"*(Ababor, 2003).

From above definitions Ababor (2003) confirmed 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.

2.1.1 Purpose of SDI

Designing an SDI system is markedly influenced by the purpose for which the system is established. Ostensibly, the purpose of an SDI system is to channel new information to points within an organization where the probability of usefulness is high (Bivona & Goldblum, 1966). The purpose of SDI is reducing user effort in selecting relevant information from a huge set of information available (Ababor, 2003).

As indicated by Rao (1993), the reason for introducing SDI service is the rapid growth of technological and social changes required for the relevant information to be instantly made available to users, researchers.

The primary purpose of SDI service is creating a usable library environment, by 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.

2.1.2 SDI Procedures

SDI systems typically consist of two major elements, information providers (library) and users. Information item distribution from provider to user proxy is based on some kind of user interest profile (O'Neil, 2001). The profile describes the types of information in which a user is interested; free-text queries are of particular to us.

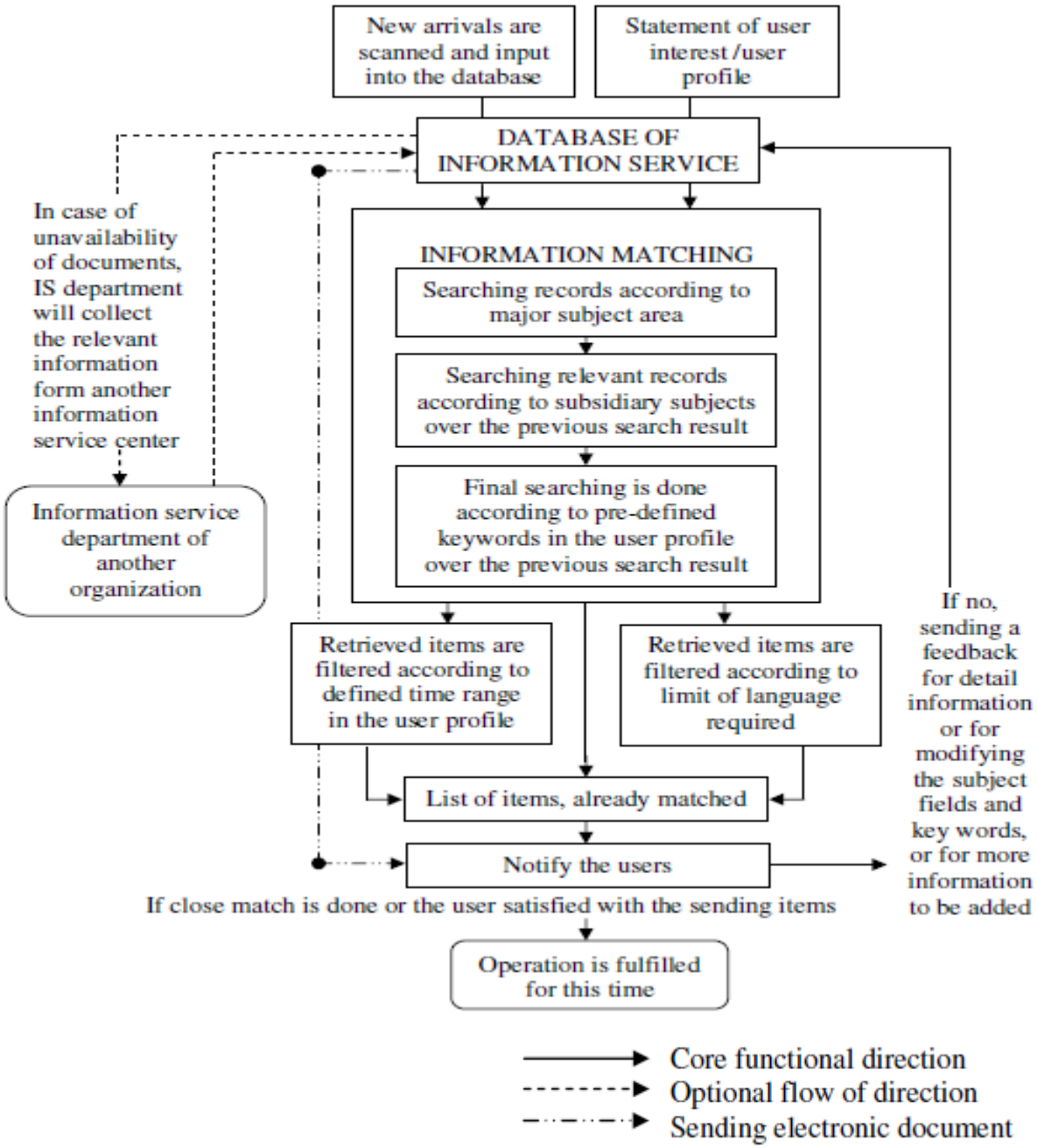


Figure 2.1 Conceptual mapping of online SDI service (Source: Hossain & Islam, 2008)

SDI service passes through three stages in which under each stage there are several steps (Hossain & Islam, 2008)

Stage one: **Scanning new arrivals and users' profile**

This function is not an internal process of SDI program. But the whole process of online SDI program is based on this recorded information.

- **Scanning the new arrivals**: Examines or scans every new document received by library in terms of subject field very clearly and widely to ensure maximum subject coverage as well as input them into the storage of the library; either in database or in file repository.
- **Scanning the users' profile** Database should contain separate functions for SDI service associated with user profile. User can open a profile on his/her personal name, then fill-up and submit it. Profile formulation can be done from computer or manually. However, in case of manual submission, the information officer should input the details of user profile into the database carefully.

Stage two: **Information matching**, which involves three steps.

Step 1: In this step, required information is to be plotted over the whole records of the library's' database for exact matching of subject matter. This function is done in three levels.

- **Level-1:** Searching to choice the major specific subject field over the whole records of the library database.
- **Level-2:** Searching to pick the subsidiary subject field (if any in the user profile) the last search result on major subject field.
- **Level-3:** Searching broadly according to pre-determined information need in the user profile, over the previous search result on related subject interest to select the appropriate items of information.

Step 2: Time range description: In this step, search procedure is performed according to required time range of information published over the past search result on required subject matter.

Step 3: language specification: In this step, search process is done parallel according to language of information over the previous search result on required subject matter.

Note that, in case the user's query is not limited to certain range of time in which 'the document published' and when no separate medium of language of information is required, then final matching function is performed directly over the previous search result and document description is prepared.

Stage three: **Notification of document to users based on his/her query for information**

This stage involves four steps:

Step 1: Sending notification to the users: The list of closely matched or relevant information is mainly sent to the concerned users in the form of bibliographical lists.

Step 2: Ultimate matching: User will try to confirm the received bibliographical items for last matching in terms of his need for information.

Step 3: Sending user's feedback to SDI system: After verification, user will send a feedback to library through SDI system. It is essential that how much information are sent or added to it. User may also ask the information officer for the distribution of selected information on descriptive form.

Step 4: Delivery of voluminous textual information: Based on user's feedback, textual information is sent in detail according to his need at any form of information dissemination (i.e. hand-to-hand, postal service, email, etc.).

2.2 Construction of User Profile

Knowing the interest of users is vital part of any information delivery. When the service to be provided is the supply of current awareness, it becomes paramount. This is because a current awareness service, by its nature, must be closely tailored to the interest of the recipients. Finding out information about the needs and interests of users is called user profile. It is something which must be undertaken inconsiderable detail (Hossain & Islam, 2008).

A user profile involves of information about user that has bearing on the user's information needs. A simple user profile is much like a query. It includes a set of key words. Such profile is originally developed for simple need of using Current Awareness Information (CAI). Away from simple user profile, an extended user profiles that is intended for the researchers contains information that is hard to correlate with documentary information but may still effect retrieval.

In order to provide personalized information to a user, the system creates and maintains a description of the type of information that the user needed to access. Modified content is retrieved based on information matching the user profile (Renganathan et al., 2013).

According to Renganathan et al. (2013), the user profile is divided into two categories static or dynamic. Static profiling is the process of receiving a user's characteristics, such as age, gender, profession, etc., via direct input from the user.

After all, an information service center (library) is always meant to be for the interests of its users. Clearly, the more you know about those users, the more easily you will be able to bring the information that they exactly need. If no effort is made to clearly understand the user's actual need, then how a successful result was delivered to concerned users.

The efficiency of an SDI system largely depends on the construction of individual or group user's profiles. The quality of user profiles has a major impact on the performance of information retrieval and filtering systems (Renganathan et al., 2013).

Therefore, a good user profile should involve the following features of information related to concerned users (Hossain & Islam, 2008):

User's identity specification: user's personal information such as name, address, contact no., e-mail, etc. is to be used.

Qualifications and experience of the users: This feature comprises the user's educational qualifications, related experience and the profession earned. It assists to gather qualitative information in perspective of user's knowledge and perceivable competences.

Subject coverage in details: This is the basic part of user profile on which the required information is analysed, retrieved and then delivered to the concerned users. Subject coverage should be wisely taken that could be explained into following segments (Hossain & Islam, 2008):

- Scope of the core subject matter: To analyze the core subject matter, it is essential to remark the broader subject of the study, related subject, title or topic of the study and a number of possible keywords that can strongly interpret the core subject area while vital information is not skipped over.
- Purpose of the study: It is an essential part in determining the volume of information to be distributed based on mode of work such as research work, article writings, and making lectures.
- Date range of publications: This option indicates that information is to be gathered which are published within this specified date range.

- Medium of language required: Information may be published in different language. Therefore, there should be an option to identify that on which medium of language information is to be delivered.
- Sources of reference: It would be very useful if the user also mention some references to bibliographic sources, which he/she considers relevant to his/her area of work.

Level of computer literacy: The option is needed to access the user's skill on computer operation. It will help to select the nature of information sources such as Information on digital source, printed hard copies, etc. as user's skill on information handling in variant sources.

Mode of distribution or delivery of information: This option is about the distribution of finally accumulated information to the targeted users which covers the type of presentation (i.e. providing only bibliographic lists of information or bibliography along with articles); form of distribution (i.e. printed hard copies, soft copies); style of delivery of information (i.e. hand-to-hand supply, delivery by postal service or delivery through e-mail).

Frequency requested in distributing information: actual interval option should be exist in user profile (i.e. weekly, monthly, etc.) the users would like to be notified about latest information on his/her interested subject interest? This option is definitely useful for the researchers and for those persons who are engaged in doing any long-term development projects.

It can be seen that the preparation and maintenance of user profile is the most significant aspect on which the whole concept of 'SDI' service involves.

2.3 Requirements for online SDI functions in database application

User profile, nature and characteristics, subject coverage, retrieval tools and techniques and specialties are some requirements for online SDI functions (Hossain & Islam, 2008).

Nature and Characteristics

The database should be text information management system operating at any network environment and linked with the library's own website.

Subject Coverage

The success of online SDI service mostly depends on scanners ability to filter the concerned documents on their correct subject matter by selecting broader and subsidiary subject interest and appropriate keywords that will lead the document on its correct core contents.

- Core subject matter of the document is to be covered by using at least one general term of subject, more than one subsidiary subject and a feasible number of keywords as to guarantee maximum coverage and use of the concerned document by maximum number of users.
- An appropriate heading with keywords should insert into database against each chapter heading of the document.
- A brief outline of useful chapter, for which keywords already setup, should also be inserting into database.
- In case of multilingual delivery of document, all the subject fields particularly should be translating into English term parallel with other terms.

Note: In fact, the utility of a multilingual retrieval system may depend abundantly on the intended body of users. Many potential users cannot read and understand a second language well enough to be benefit from multilingual systems.

Specialties

- There may be a terminological dictionary (thesaurus) of database for searching information with exact spelling of search key word in which all the terms are to be arranged on a logical relationship.
- All the nearest similarly spelling words were appeared as tool-tips during the time of typing a term in free-text searching. This is to ensure the correct spelling of the search key word as well as giving information about related subjects' terms in advance.
- There may also an automated indexing system in which whole terms are linked with the related sources of information. In this system, all terms are saved automatically into the database and arranged in an alphabetic order whenever a new term is added into any field of keywords.

- An elementary option in support of online SDI service is, the opportunity of saving the search profile so that it may be executed again later. The search results may also be saved under appropriate subject heading/headings for general reference purposes.

2.4 Information Filtering

Most authors such as (Belkin& Croft, 1992; Renganathan et al., 2013), described that information filtering and information retrieval share many similar features with few difference. However, to be sure Information filtering is SDI (Belkin & Croft, 1992; Porcel, Moreno& Herrera-Viedma, 2009).

2.4.1 Features of Information Filtering

Belkin & Croft (1992) putted features of information filtering as follows.

An information filtering system is developed for unstructured or semi-structured data. This is different from a typical database application that includes very structured data. The notion of structure being used here is not only that the data conforms to a format such as a record type, but also that the fields of the records contain simple data types with well-defined meanings. For example, to define a database type for a complex document, such as a journal article, picture and table components of that type are much less well-defined than a typical component of database, such as the salary attribute of employee record. Email messages are an example of semi-structured data in that they have well-defined header fields such as address and subject and an unstructured text body.

IF systems deal primarily with textual information. It is, however, more general than textual information and should include part of multimedia information systems such as images, voice, and video. None of these data types is handled well by structural database systems, and all have meanings that are problematic to denote.

Filtering systems involve large amounts of data from streams of incoming data, either being broadcast by remote sources or sent directly by other sources (email). Typical applications would deal with gigabytes of text, or much larger amounts of other format.

Filtering has also been used to describe the process of searching information from remote databases, in which case the received data is the result of the database searches. The designers of

systems that generate “intelligent agents” for searching remote, heterogeneous databases also use this scenario.

Filtering is based on statements of individual or group information need, often called profiles. Such profiles typically represent long-term subject interests.

Filtering is often meant to imply the exclusion of data from an incoming stream, rather than finding information in that stream. In the first case, the users see what is absent after the data is detached; in the second case, they see the data that is extracted. An example of the first method is an email filter designed to remove "junk" mail. Note that this means profiles express both what people want and what they do not want.

Many of these features are nearly the same as those found in a variety of other text-based information systems (IR, Text routing, Categorization) (Belkin& Croft, 1992).

2.4.2 Information Filtering vs Information retrieval

As Belkin & Croft(1992) and Renganathan et al. (2013) noted, the major difference btween Information filtering and Information Retreval is the static nature of the categories, when compared to profiles.

IR is normally concerned with single uses of the system, by a person with a one-time goal and one-time query, while information filtering is facilitated with frequent uses of the system, by a person or persons with long-term goals or interests.

IR recognizes inherent problems in the adequacy of queries as representations of information needs, filtering take up that profiles can be exact specifications of information interests.

IR is concerned with the collection and organization of texts, while filtering is focused on the distribution of texts to groups or individuals.

IR is typically concerned with the selection oftexts from a relatively static database, IF is mostly concerned with selection or elimination of texts from adynamic data stream.

IR is concerned with responding to the user’s communication with texts within a single information-seeking episode, while filtering is concerned with long-term changes overa series of information-seeking episodes. Notwithstanding these distinctions taking into account the models of IR and separating, there appear to be some other, logical contrasts that may likewise be

significant to research interests. These come from differences in the social and/or practical circumstances with which IR and filtering have been concerned. Such differences could be categorized related with texts, users, and general environment of concern to each.

Text related issues. For information filtering, the time liness of a content is frequently of abrogating essentialness. For IR, this has regularly not been the case.

User related issues. IR has studied by well-defined user groups, in a precise specific areas, largely in science and technology. These users have almost always been highly motivated in the irinformation-seeking behaviors. However, filtering is often concerned with very undefined user communities, such as people looking for entertainment in their homes, and with highly varied areas. Also, motivation in the filtering environment is some times difficult to be assumed.

Environmental issues. Here, the most salient differenceseems to be that filtering is extremely concerned, in different angles, with issues of confidentiality; IR has paid almost no attention to this kind of problem.

Renganathan et al. (2013) summurized the difference between information Retreval and information filtering as below table

Table 2.1: Information retrieval *vs* filtering system.

	Informatio Retereval	**Information Filtering**
Information need	Dynamic	Static
Information source	Static	Dynamic
User profile	Not necessary	Essential
Scope	Generalized	Specific
Information seeking Behavior	Short term	Long term
User Query	Brief	Description or explanation of the information
User interaction with the system	Single information seeking episodes	Series of information seeking episodes

Even though, the goal of displaying relevant information is Common to IR and IF, but they differ in the following aspects (Hanani, Shapira, &Shoval, 2001):

- Frequency of use: IR systems are developed for ad-hoc use of a one-time user, to fulfill a one-time user interest. IF systems are developed for long term users with long term information needs (subject interests), and for repetitive usage.
- Representation of information needs- in IR systems, user needs are expressed as queries. In IF systems, long-term user information needs are labeled in user profiles.
- Goal- IR systems select from databases relevant data items (documents) that match a query. IF systems remove, irrelevant data from incoming streams of data items, or collect and disseminate relevant data items from some sources, in accordance with a user's profile.
- Database- IR systems deal with relatively constant databases. IF systems deal with dynamic data (like e-mail messages).
- Type of users- IR systems serves users who are not known to the system; anyone who has access to the system may pose a query. Users of IF systems are known to the system from users database; the system has a model of the user, usually recorded in the form of a user profile.
- Scope of system- IF systems are concerned with social issues like user modeling and confidentiality that are most of the time no anxiety to IR systems.

2.4.3 Types of Information filtering

Based on the data acquisition, there are three major approaches for information filtering: Content-based filtering, collaborative filtering and knowledge-based information filtering. For better performance, also researchers combine two of these three approaches and make the fourth type of information filtering called hybrid information filtering.

Content-based filtering

Content Based filtering system recommends an item by matching the document profile with the user profile, using information retrieval approaches such as Term Frequency and Inverse Document frequency (TF-IDF). User features are gathered over time and stored automatically based upon a user's prior feedback and choices. The system uses item-to-user correlation in recommending the item to the user. The system starts with the process of collecting the content

details about the item, such as treatments, symptoms etc. for disease related item and author, publisher etc. for the book items. Then, the system asks the user to rate the items. At the end, system matches unrated item with the user profile and assign score value to the unrated item and user is recommended with items ranked according to the scores assigned (Spiegel, 2009; Renganathan, et al., 2013; Meteren & Someren, 2000).

Content-based information filtering systems are not affected by the cold start problem and new user problem, as the content-based information filtering system depends on the individual user's information needs. Content-based information filtering systems are not suitable for items such as images, audio, video. So, multimedia documents must be tagged with a semantic description of the resource, which the process is time consuming. Content-based filtering methods cannot filter documents based on quality and relevance (Renganathan et al., 2013).

Collaborative filtering

As Renganathan et al. (2013), Collaborative filtering systems screens information based on the interests of the user past history, and the ratings of other users with similar interests. It is broadly used in many recommender systems, especially in e-business applications. One example of such system are Amazon.com and e-Bay, where a user's past shopping history is used to make recommendations for related new products.

Collaborative filtering system involves the computation of similarity between user interests (Spiegel, 2009). Similarity between the users interest are calculated using different methods such as Pearson correlation coefficient (Ababor, 2003). The system collects the ratings of each item from different users explicitly or through their browser behavior, and then calculates the similarity between the ratings of the users. The ratings can be explicit on a numeric scale, or implicit such as purchases, clicks and mouse movement. Then, the users are grouped based on correlation between them and future items are recommended to the user based on the recommendation of other users in the group (Renganathan et al., 2013)

Consider a group of users U_1 through Un and items I_1 through Im, which is presented in table 2.2 below.

Table 2.2 rating given by the users on different items (Source: Renganathan et al., 2013)

	I1	**I2**	**I3**	**…**	**Im**
U1	1	4	4		4
U2	1	3	4		3
U3	2	4	3		5
U4	2	4	3		
U5	1	4			4
…					
Un	3	4	1		4

For example, if similarity rating between the user U_1 and U_5 is high, then user U_1 and U_5 can be grouped and new items was recommended to each user based on the other user's interest. Here, item $I_3$was notified to the user U_5, as a new item based on the high rating given by the other user in the group U_1. Similarly, item Im was recommended to User U_4 based on the rating of other user U_3.

The collaborative systems can be used to filter all types of items, including the multimedia items. It suffers from the cold-start problem and early rater problem. It includes the issues of filtering a new item, if any one of users does not rate it yet. This system also suffers when data are sparse, which makes the recommendation difficult, as there are rare common items present in calculating the similarity measures (Renganathan et al., 2013; Wang et al., 2006).

Knowledge-based filtering system

A third type of filtering system is one that uses knowledge about users and products to pursue a knowledge-based approach to generate a recommendation, reasoning about what products meet the user's requirements (Burke, 1999). It is preference based filtering system and so, it suggests products based on inferences about a user's needs and preferences.

Hybrid filtering systems

The fourth filtering system is hybrid filtering system. The hybrid filtering systems combines features of two filtering systems or more techniques and avoid some shortcomings (Spiegel, 2009). For example, in the combination of content-based and collaborative filtering, the hybrid system overcomes the problem of cold start and early rater problem by using the content based in the initial stage. In the subsequent stages, features of collaborative filtering systems is used, which benefits the system to recommend all types of items, including multimedia items and overwhelms the problem related to content based filtering techniques (Renganathan et al 2013).

In general, hybrid recommender is a system that combines different recommendation techniques together to achieve a synergy between them (Spiegel, 2009). Although there exist a numerous approaches that are practical to merge (i.e. Collaborative, Content-based, Demographic and Knowledge-based Recommender), Spiegel mainly focused on the combination of CF and CBF techniques.

2.4.4 Information Filtering Models

Information Filtering is introduced based on Information Retrieval models (Aberer, 2006; Shuda, Jiangping, & Riu, 2009). Therefore, the widely used IR Models such as Vector space model (TFIDF term weighting and Latent Sematic Analysis (LSA) model are also used in designing information filtring system.

- **String-matching model**

In the string-matching model, the user specifies his/her information needs by a string of words. A document would match the information need of a user if the user-specified string found in the document. This model is one of the earliest and simplest approaches. String-matching model is less able to match the documents that need contextual and experiential knowledge, and also it suffers from the problems of homonymy (words are spelled same, but have different context), synonymy (the same context can be expressed by different words), polysemy (different contexts can be expressed by the same words) and bad response time (Renganathan et al., 2013).

String-matching model works by selecting documents, which consists of term/s that existed in the user's information need. If the document contains single word of user's information need, that document is recommended for user without considering the importance of that term.

- **Vector space model**

The Vector Space Model (VSM) or term vector model is an algebraic model representing textual information as a vector which is used for Information Filtering, Information Retrieval, indexing and relevancy rankings (Polyvyanyy & Kuropka, 2007). VSM is a space where text is represented as a vector of numbers instead of its original textual representation; the VSM represents the features extracted (words/terms) from the document.

In the VSM, documents are identified by terms. i.e the meaning of a document is conveyed by the words used in that document (Clark, 2014). A document D is represented as a vector of dimension m, where m is the total number of terms used to identify content. Each term is given a weight that signfies its statistical importance (Yan & Garcia-molina, n.d.).To find relevant documens for a given profile (query), VSM proceedes three stages: first Document indexing; map documents and profile into term-document vector space and content bearing terms are extracted, Second weighting of indexed terms; Profiles and documents are represented as weighted vectors W_{ij} where W_{ij} is wieght of term i in document j and third similarity between user information need and indexed document is computed.

During inexing**,** considering document preprocessing such as Tokenization (to split the text into individual words), stopword removal (which is descarding most frequently occured words that are found in all documents), Token normalization (which depending on the task as case folding (discarding information about letter casing)) and stemming (which depending on bringing tokens into its bases or roots by removing prefix or suffix from it) are fundamental processes (Polyvyanyy & Kuropka, 2007; Rehurek, 2011).

Term weighting for the vector space model is handled statistically. Weighting schemes are (mostly empirical) attempts to model the underlying relationships between the importance of individual dimensions, token frequency distribution and varying document lengths (Rehurek, 2011).

There are some factors of term weighting (Polyvyanyy& Kuropka, 2007): Term Frequency (TF), Document Frequency (DF), Inverse Document Frequency (IDF) and Term Frequency-Inverted Document Frequency (TF-IDF)

Term Frequency (TF) is count of term in a document. i.e. TF measures the number of times a term (word) occurs in a document (Vembunarayanan, 2013). Documents that have use many terms get more matches than short documents. So long, documents have unfair advantage on short documents. Therefore, to remove this effect it is important to normalize the document based on its size.

$$tf_{ij} = \frac{fij}{\max(fik)} \quad \text{2.1}$$

Inverse Document Frequency (IDF) is measures rarity of the term and therefore, it is measure of the general importance of the term in collection. In the Boolean model all terms are considered equally important. In fact, certain terms that exist too frequently in whole documents have little power in determining the relevance. We need a way to weigh down the effects of too frequent terms. Also the terms that occur less in the document can be more relevant. We need a way to increase the weight of less frequently occurring terms. This is the core factor to come up with vector space model. We use logarithm function to measure IDF of term i.

$$IDF = \log_2(N/dfi) \quad \text{2.2}$$

Where, N is total number of documents and df_i is total number of documents term i found in.

The inverse document frequency (IDF) assumes that the importance of a term is proportional with the number of documents the term appears in (Polyvyanyy& Kuropka, 2007).

Term Frequency-Inverted Document Frequency (TF-IDF) is product of normalized term frequency and inverse document frequency. A weight of term i in document j is its TF-IDF value.

$$\rightarrow tf_i * idf_i \quad \text{2.3}$$

Similarity measure is a function that computes the degree of similarity or distance between document vector and query vector. Similarity is word overlap between user information need

and document. The most common similarity measure for the Vector Space Model is cosine similarity, which measures cosine of the angle between two vectors (profile and document) in the vector space (Rehurek, 2011). Cosine similarity is normalized inner product of profile vector and document vector. That means it is a division of inner product by product of magnitude of profile (query) and document (item).

$$sim(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{\left|\vec{d}_j\right\|\vec{q}\,|} = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2}\sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \quad \ldots\ldots\ldots\ldots\ldots\ldots 2.4$$

- **Latent Semantic Analysis (LSA)**

As a base of Vector space model is a term frequency (term weighing), a fundamental deficiency of a model is that information needs are often not the same as words those by which documents has been indexed. That means the issues of synonymy and polysemy are not considered. Synonymy in a very general sense is to describe the fact that there are many ways to refer to the same object. Users in variety of contexts or with different needs, knowledge, or linguistic habits will express the same information using different words. Indeed, we have found that the degree of variability in key term usage is much greater than is commonly suspected. The prevalence of synonyms tends to decrease the "recall" performance of filtering systems. Polysemy refers to the general fact that most words have more than one unique meaning (homography). In different settings or when used by different people, the same word takes on varying referential significance. Thus, the use of a term in user's information need does not necessarily mean that a document containing or labeled by the same term is of useful. Polysemy is one factor results in poor "precision" (Deerwester, Dumais, Furnas, & Landauer, 1990; Rosario, 2000).

Therefore, Latent semantic analysis (LSA) also called Latent semantic indexing (LSI) is proposed by Deerwester, et al in 1990 to cut-off this issues. Its main idea is to exploit term co-occurrence to derive a set of latent concepts; words that frequently occur together are assumed be more semantically associated (Rehurek, 2011). This is in accordance with the statistical semantic hypothesis, as it directly models the relationship between words based on the contexts that they share. LSA analyzes term co-occurrence of higher orders, so that it is able to incorporate the

relationship of words A, B which only co-occur in documents through a difference word, C, and never appear in the same document together directly. In the same way, LSA is able to de-emphasize a connection between two co-occurring terms that frequently co-occur with many other terms at the same time.

LSA uses one of the most widely used matrix decomposition algorithms Singular Value Decomposition (SVD). Latent semantic indexing is the application of a specific mathematical technique, called Singular Value Decomposition or SVD, to a term-by-document matrix. SVD (and hence LSI) is a least-squares method(Rosario, 2000). In the context of LSA, documents represented in a space of much lower dimensionality, the truncated target rank <<the number of features (matrix rows) (k<<m), both saving resources and getting rid of data noise at the same time. For this reason, SVD can be viewed both as a dimensionality reduction and a noise reduction process, improving efficiency and efficacy at the same time(Rosario, 2000; Rehurek, 2011).

However, LSA also suffers from several shortcomings (Rehurek, 2011; Rosario, 2000):

- Choosing the optimal value for the k parameter (the latent space dimensionality) is not obvious. In IR practice, the value is typically set to several hundred, but depends on the application as well as the structure of the input corpus
- Topics are not interpretable. In other words, by looking at the m-dimensional vector of a particular topic t_i, $i \leq k$, it can be hard to assign a human label to the theme connecting the highest scoring terms
- Using Latent Semantic Indexing vectors, we can no longer take advantage of the fact that each term occurs in a limited number of documents, which explanations for the sparse nature of the term by document matrix.
- With LSI, the query must be compared to every document in the collection. So the efficiency is slow down.

2.5 Related Works

From reviewed literatures, the researches realized that there were limited works done in the area of SDI for academic libraries of Ethiopia. However, internationally there are studies with the aim of recommending articles from digital libraries as well as from the web.

Ababor (2003) proposed a collaborative filtering agent for document recommendation in SDI system for International Livestock Research Institute (ILRI) SDI service. Ababor did experiment based on the neighborhood algorithm and used Pearson correlation coefficient as a similarity measurement.

$$w_{a,u} = \frac{\sum_{i=1}^{m}\left[\left(r_{a,i} - \overline{r_a}\right)\left(r_{u,i} - \overline{r_u}\right)\right]}{\sqrt{\sum_{i=1}^{m}\left(r_{a,i} - \overline{r_a}\right)^2 \sum_{i=1}^{m}\left(r_{u,i} - \overline{r_u}\right)^2}} \quad \text{.....2.5}$$

Pearson correlation equation 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; is $\overline{r_a}$ 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, is $\overline{r_u}$ 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 she takes user_ id 10 as an active user, then user 10 is our a and our u's was all the other users one by one. That means she first calculated 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.

Ababor chooses number of best neighbors over correlation threshold 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.

In her research, Ababor evaluated performance of collaborative filtering algorithms by using coverage and precision.

Coverage is usually computed as a percentage of items for which the system was able to provide a recommendation.

$$\text{Coverage} = \frac{number\ of\ items\ recommended}{total\ number\ of\ items} \quad \ldots\ldots\ldots\ldots\ldots\ldots 2.6$$

Precision is defined as the ratio of relevant documents selected to number of documents selected. It is computed as $P = \frac{Nrs}{Ns}$ 2.7

Where N_{rs} is relevant document selected and N_s is sum of both relevant and irrelevant document selected.

Ababor realized that the highest coverage of the system was registered when all other users in the database are considered as best neighbors. The coverage becomes very low when the size of neighborhood drops to less number.

Porcel et al. (2009) developed SDI or information filtering for University Digital Library (UDL) by using fuzzy linguistic model recommender system to achieve major advances in the activities of UDL in order to improve their performance.

Their system is oriented to researchers and it recommends two types of resources: specialized resources of the user research area and complementary resources in order to include resources of related areas that could be important to discover collaboration possibilities with other researchers and to form multi-disciplinar groups.

Porcel, et al (2012) again developed a hybrid recommender system for the selective dissemination of research resources in a Technology Transfer Office, University of Granada, Spain in the management of research resources. This system uses a fuzzy linguistic modeling to represent the qualitative information presented in the system communication processes Particularly, they use a multi-granular fuzzy linguistic modeling that provides greater flexibility in the user-system interaction, which turns to be stimulating and useful characteristic.

The system implemented based in a switching hybrid approach which shifts between a content-based recommendation approach and a collaborative one to share the user individual experienceand social wisdom.They used MovieLens data sets to develop the offline experiments.

Altinel & Franklin (2000) have developed different index organizations and search algorithms for performing efficient filtering of XML documents for significant information dissemination systems. They developed a document filtering system, named *XFilter* that provides highly efficient matching of XML documents to large numbers of user profiles. In XFilter, user interests are represented as queries using the XPath language. XPath is used to select entire documents rather than parts of documents. That is, they treat an XPath expression as a predicate applied to documents. If the XPath expression relates at least one element of a document, then the document satisfies the expression. The XFilter engine uses a sophisticated index structure; modified Finite State Machine (FSM) approach to quickly locate and examine relevant profiles. They described these structures along with an event-based filtering algorithm which drives the process of checking for matching profiles in the Index using XML parser.
By converting XPath queries into a Finite State Machine representation, XFilter is able to:

- Handle arbitrary regular expressions in queries
- Efficiently check element ordering and evaluate filters in queries, and
- Cope with the semi-structured nature of XML documents.

They reported XFilter as effective for different document, workload and scale scenarios, which makes it appropriate to use in Internet-scale SDI systems.

The other work related to this study is the one that is done by Morales-Del-Castillo, et al (2013). They developed a Semantic Model of Selective Dissemination of Information for Digital Libraries. Their model has four basic components that made up the system: Thesaurus (enables organizing the most relevant ideas in a specific domain, defining the semantic relations established between them, such as equivalence, hierarchical, and associative relations), User profiles (structured representations that contain personal data and interests) with which agents can operate to customize the SDI service and generated the moment the user is registered in the system), RSS feeds (create current-awareness bulletin) and Recommendation log file (Each document in the repository has an associated recommendation log file in RDF that encompasses

the listing of evaluations allocated to that resource by different users since the resource was added to the system).

To show contribution of the work, it is better to compare the current study with some others related works that are experimented in the related environment.

Ababor, 2003 Applied Collaborative Filtering for Document Recommendation in SDI system of International Livestock Research Institute's (ILRI). Her main objective is to investigate the possibility of applying a collaborative filtering agent for making document recommendation in SDI systems. The best performance result that her system achieved is almost 55% of precision. She said, "Small neighboring size performs well in most cases" that best performance is 55% of precision. To improve performance of the system, she recommended application of hybrid filtering in SDI system.

Therefore, we have to compare our work again with other SDI system developed by the application of hybrid filtering. So we would like to take the work of (Porcel et al., 2012) published under Journal of Information Sciences in 2012. The main objective of the work is to help the Technology Transfer Office staff in the dissemination of research resources interesting for the users. Porcel et al. (2012) achieves 67.42% precision and 69.03% recall. Their system performs better result than Ababor's work.

Both works of Ababor (2003) and Porcel et al. (2012) is based on rating of items by users while in the current study, no any action on item is required from users since users are express their information need without knowing source of items . Promising result 70.6%, 78.76% and 73.76% for recall, precision and F-measure respectively were found in the current work than previous ones. The current system also conducted user acceptance testing 95% which is not assigned by prior researchers.

In table 2.3 below, summary of the different related works with current study are presented.

Table 2.3 comparison of related work to current work

Author	Title	approach	Evaluation parameter	dataset
J.M. Morales-del-Castillo (2013)	A Semantic Model of Selective Dissemination of Information for Digital Libraries	Semantic modeling	N/A	Digital repository
C. Porcel et al (2012)	A hybrid recommender system for the selective dissemination of research resources in a Technology Transfer Office	Hybrid IF	P and R 67.42% P and 69.03% R	MovieLens
Zehara Zinab Ababor (2003)	Application of Collaborative Filtering Agent for Document Recommendation in SDI System	Collaborative IF	Coverage and precision (55% P)	Bibliographic database records
Mehmet Altınel and Michael J. Franklin (2000)	Efficient Filtering of XML Documents for Selective Dissemination of Information	Modified FSM	N/A	XML documents

CHAPTER THREE

CONCEPTUAL MODELING, DATA AND METHODS

3.1 Conceptual modeling of the SDI system

The two main components of library are documents and users. So having a record of these components is fundamental task of SDI service. Because, data used to develop SDI service are generated from these components. Therefore, beginning from these components, the conceptual model of proposed SDI system is sketched in Figure 3.1 below.

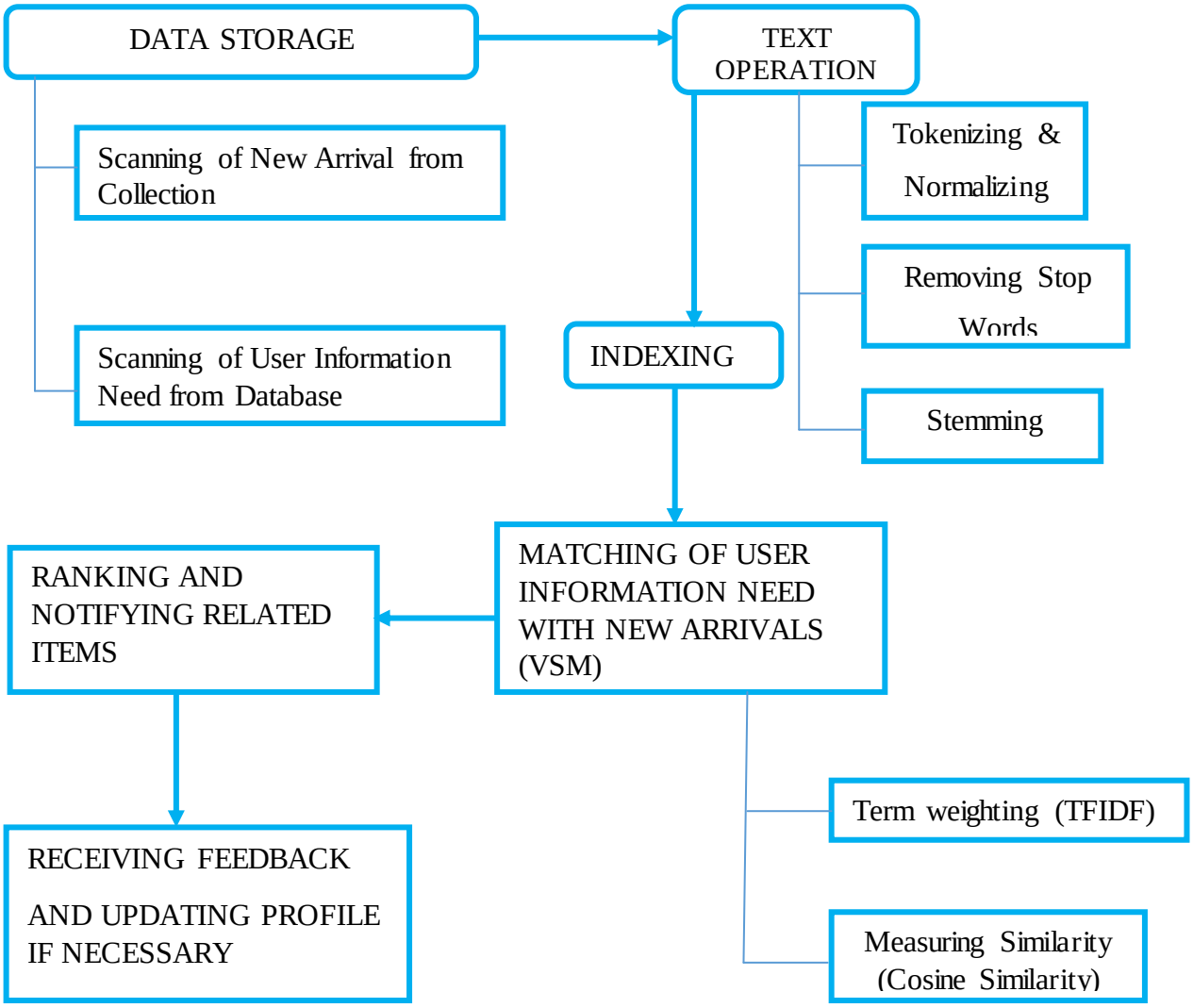


Figure 3.1 Conceptual modeling of SDI system

The **first stage** in SDI system is having documented information about items and users. This is the initial stage of SDI service. Tasks in this stage are related to scanning of new arrived items from collection of received books and inputting them to SDI service. In addition, recording and storing information of users' profile into a database.

Scanning new arrival books: Identifies new file containing lists of arrived books from received books collection. There may be many files in the file system and therefore, it is necessary to identify recently submitted file containing bibliographic lists of books.

Scanning the users' profile: User can open profile on his /her name, fill-up other information as required by service provider and submit it. Profile opening and submission function can be also done from PC. This means data acquisition about users can be done automatically and manually. But in the case of manual submission, the SDI operator should input the details of user profile into the database carefully.

In manual knowledge acquisition, knowledge can be acquired directly from user or from other source describing the user. For example, in this study we provide users' profile database after we get the information about most of user from Jimma University website as well as from some of users directly.

For training purpose, profile opening was done and the researcher directly filled records into database. However, after SDI system is designed, user was able to open profile on his/her own and submit to SDI system.

After user registration is accomplished whether manual or automatic, user information need should be identified to be input for next stage. In this study major subject interest and subsidiary subject interest are information need identified. Because other attributes are not necessary to be extracted.

Stage Two:. Text operation. Under this stage, tasks such as tokenization, normalization, stop word removal and stemming are take place.

Stage three: Indexing. After knowledge or data are acquired, and text operation is processed, the next task is indexing documents and user information need to make the recommendation of item fast

Stage four: The operation in this stage is information matching. In this stage, required information is to be surveyed over the whole documents of new file identified in the stage one for best matching of information need (subject matter). So, tasks under taken here are term weighting and similarity measurement.

Stage five: Recommendation of related documents to user. The final lists of closely related or relevant information are recommended for user.

Stage six: Feedback and profile updating. In this stage user is asked whether he/she is satisfied with the documented recommended to him/her. Based on his/her satisfaction level user may or may not update his/her profile. Whatever the satisfaction level, the system registers user satisfaction. For short, user can update his/her user information need if he/she is not satisfied with the document forwarded to him/her.

3.2 Dataset preparation and preprocessing

For the development of SDI service, dataset can be prepared from sources, such as data warehouse, database or full text repository (Morales-Del-Castillo, et al, 2013) and a flat file or spreadsheet. In this study, a dataset available in spreadsheet (.xlsx) file is used.

However, it is important to convert file to format that is compatible to our system development tool (python in our case). As it is known, internal operation of SDI service is based on text file. Therefore, data format that supports text operation is a prerequisite for the preparation of SDI service. In this work dataset used was in .CSV (comma separate value) file format. File, which is, referred to us a flat file, stores tabular data (numbers and text) in plain text. Each line of the file is a data record (one document in this work). Each record consists of one or more fields, separated by commas[3]. Because, CSV file has no structured interrelationship. The term is frequently used to describe a text document from which all word processing or other structure characters or markup have been removed[4].

[3]https://en.wikipedia.org/wiki/Comma-separated_values

[4]http://searchsqlserver.techtarget.com/definition/comma-separated-values-file

Besides having CSV file as a reason to support processing of text documents, python has also separate module (package) that supports importing of CSV file. Therefore, it is possible to read CSV file from directory and write to CSV file with package called CSV.

Dataset Cleaning

Data should be cleaned before using it for SDI. Because, we do not always have control over the format and type of data that we import from an external data sources. Misspelled words, persistent trailing spaces, unwanted prefixes, improper cases, data redundancies and nonprinting characters make a bad first impression. That is not even a complete list of ways a data can get unclean. Therefore, before the data is used in the system development tools, there is often a need to clean it up. This process of cleaning up the data is called data preprocessing.

There are a number of data preprocessing techniques. Among these, data cleaning can be applied to remove noise and correct discrepancies in the data while data reduction can reduce the data size by aggregating, eliminating redundant features.

Dataset prepared for this study passed through two steps of data cleaning; first removing duplicated rows, and then removing nonprinting characters.

Sometimes unintentionally, the same record may exist more than one time especially in the library when materials are acquired. This is because different departments may ask the same document for purchase since source of data for acquisition is from department. Due to data size, library suffers to identify duplicated documents overall. Therefore, those duplicated items should be removed to increase the performance of SDI service. For example, if our system recommends document based on threshold value, let us say top ten and two of the documents are duplicated, user is recommended only eighth document. As a result, user may hate our system and from system wise, user lost two important documents that should be recommended. For this study, five thousand (5000) lists of purchased bibliographic information of books were gathered from library acquisition. Among these, one hundred seventeen (117) were duplicated and removed from the file. Therefore, four thousand Eight Hundred Eighty Three (4883) records are cleaned dataset that is used for this study.

After duplicated rows are removed, the file was imported. During this time, the researcher phased another problem regarding dataset. This is nonprinting characters was detected and the developed system get complicated to bring out the result during run time. One main drawback of CSV file is that the interpreter/compiler gets confused when the field data may also contain commas (,) or even apostrophes (‘). When such characters occur in the file (field), for example, application such as MySQL and Python understood in their encoded characters and refused to interpret the file. So a CSV file implementation may include escape characters like back slash (\). However, it is not difficult to guess how it is challenging to find out the occurrence of comma (,) or apostrophes (‘) in such big size records. In python the task is most challenging since it does not show what is detected unless giving error report “unknown character is detected at some position”. But MySQL has some advantage on python to detect such characters in their encoding characters. MySQL highlights and shows encoded characters, which prevent a record to be imported in the table like this: x92, xA0, xFC and so on. Thus we created book table in JULS database and imported CSV file into table and pick out highlighted characters one by one manually. After cleaning highlighted characters, table accepts and imported records and book table is filled with bibliographic book record. From book table we exported CSV file to file system again. Through these steps we cleaned our dataset. Therefore, final dataset used for this study is which is exported from book table as CSV file.

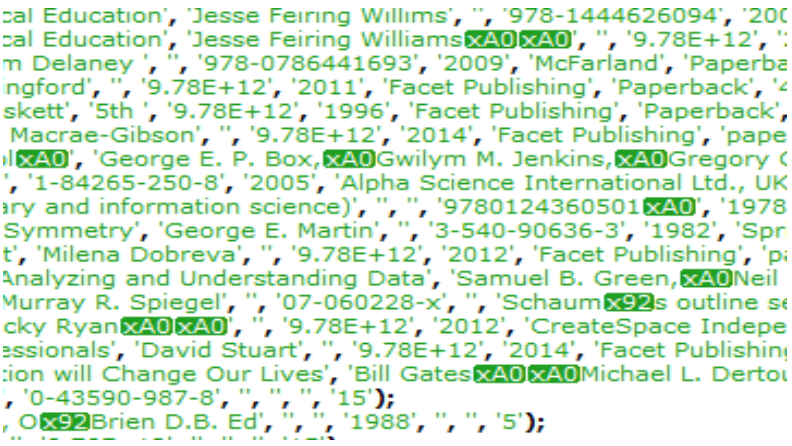


Figure 3.2 Examples of nonprinting characters (highlighted one)

Identifying new file

The file should be saved on the disk from time to time as soon as books are received and data preprocess has been finished. That means, we have different files on the disk. So the question has to raise here is should we consider all file from document collection (directory)? Remember our objective is to recommend users newly arrived documents. So the researcher decided to use latest file added to the directory of file containing arrived books separately as an input to recommend items (records) from it.

This has an advantage over other approach of information filtering. For example, Collaborative filtering suffers from problem of user information item matrix sparse as information items exceeds what users absorb and system performance is lower with increment of users and information sources (Wang et al., 2006). That means in collaborative filtering, data sources are considered from the beginning. In this work, the researcher would like to use only single file which is relatively latest one (recently added file) to reduce data sparse.

Attribute selection

Once file to be used is identified, the major issue is thinking which attribute of a file can represent whole document to be used as an input for SDI system. Dataset for this study has eight attributes (Title, Author, Edition, ISBN, Pub-Date, Publisher, Binding and Quantity). We think that from these attributes someone can guess, as Title can be representative of a document or record. This is because it is difficult to relate user information need with other attributes except title since user information need is expressed in full text. So from our data set, only title contains words that can be indexed since it is also expressed in full text. But other attributes like edition, ISBN and Quantity are not necessary to be indexed since they are expressed in number while other attributes like Pub-Date, Publisher and Binding have repeated values. Author may not be English word and it is not convenient to index them.

Therefore, even though all attributes are displayed as an output or documents are recommended with all attributes, only title was used as input for this study.

3.3 User Profile Formulation and subject interest normalization

As discussed earlier, the development of SDI service needs user profile. To store information about users we developed database. We acquired all user information from Jimma University web site except User ID. We set the User ID by ourselves. As we collected user information from web site, we registered using Microsoft Excel by adding User ID to collected information and saved it as CSV file. Then we imported that CSV file to user table. The researcher used MySQL 5.5 database management system specifically MySQL workbench 6.3 which has a graphical user interface. By using this GUI, we could easily import records from CSV file to database.

Important notation should be taken here is, like dataset, user information need should be normalized since information need can be considered as small set of documents. From web site user subject interest is not set structurally. In this research, the researcher observed that user subject interest is ranged from minimum of one statement to maximum of eight statements. That means there are users who have one statement of subject interest may be one word and users who have eight statements of subject interest. For this reason, we split subject interest into MajorSubjectInterest and SubsidiarySubjectInterest only for the sake of having reduced data length in the user table. SubsidiarySubjectInterest field is allowed null value because if user subject interest is not long MajorSubjectInterest field is enough to store subject interest.

So, how it is possible to do similarity between these huge number of user subject interests and documents? To answer this question, the researcher should search for mechanism which is help to have subject interest in a possible way to relate with documents. That mechanism is splitting user information need into statements and appending to list. See how it is implemented under chapter four (4).

We added semicolon (;) between statements if subject interests are more than one to separate one statement from other statement. By the way, this is at the time of registering user profile to excel or directly to user database. Users were used comma (,) to separate between statements that may also be used in the one statement to separate words. That why we used semicolon than comma.

3.4 Text operation

Once data set and user information need are provided, Like other systems such as information retrieval and Natural Language processing (NLP) (Aberer & De, 2006), the first step in the development of SDI system is generating structured representations of information items (documents) and generating structured representations of user information need with the process of text extraction. This means texts are extracted from file specifically from list of titles and from user information need of user profile.

After text is extracted, text operation is applied on it. By the way, text operation is broad term representing text preprocessing which incorporates tokenization, stop word removal, text normalization and stemming.

Not all words in a document are equally significant to represent the contents/meanings of a document. Therefore, it is significant to preprocess the text of a document in a collection to be used as index terms. Preprocessing is the process of controlling the size of the vocabulary or the number of distinct words used as index terms.

For text operation the researcher used Natural language Toolkit (NLTK) version 3.2 and nltk-data. NLTK was originally developed in 2001 as part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania. It has now been serves as the basis of many research projects (Bird, Klein, & Loper, 2009).

Figure 3.3 depicts the snapshot lists of the most important NLTK modules.

Language processing task	NLTK modules	Functionality
Accessing corpora	nltk.corpus	Standardized interfaces to corpora and lexicons
String processing	nltk.tokenize, nltk.stem	Tokenizers, sentence tokenizers, stemmers
Collocation discovery	nltk.collocations	t-test, chi-squared, point-wise mutual information
Part-of-speech tagging	nltk.tag	n-gram, backoff, Brill, HMM, TnT
Classification	nltk.classify, nltk.cluster	Decision tree, maximum entropy, naive Bayes, EM, k-means
Chunking	nltk.chunk	Regular expression, n-gram, named entity
Parsing	nltk.parse	Chart, feature-based, unification, probabilistic, dependency
Semantic interpretation	nltk.sem, nltk.inference	Lambda calculus, first-order logic, model checking
Evaluation metrics	nltk.metrics	Precision, recall, agreement coefficients
Probability and estimation	nltk.probability	Frequency distributions, smoothed probability distributions
Applications	nltk.app, nltk.chat	Graphical concordancer, parsers, WordNet browser, chatbots

Figure 3.3 Language processing tasks and corresponding NLTK modules with example functionality (source: Bird, Klein, & Loper, 2009)

Therefore, the Natural Language Toolkit (NLTK) is an application or package used for creating Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. Among these language processing tasks listed on figure 3.2, the researcher used the module to accessing corpora (to access English stop words for stop words removal) and to text processing (tokenization, token normalization, and stemming).

NLTK 3.2 is downloaded from, https://pypi.python.org/pypi/nltk. After NLTK 3.2 was installed, nltk_data was downloaded separately.

Nltk-data contains the linguistic corpora (stop words) that are analyzed and text processing algorithms (Tokenization and stemming (snowball stemmer)) developed using python.

Tokenization is the process of splitting documents into individual words after a specific delimiter (white space in our case (Robin, 2009). In this work, tokenization process is used to split titles into words. Then tokens are normalized. Token normalization involves the process of

handling different writing system such as case folding (lower case in this work). Also there should be punctuations mark to be removed from all words to make the same word with different punctuations marks equally understandable in similar way with no distinction of punctuation mark linked to it. To remove punctuations from tokens, Natural Language Toolkit (NLTK) Tokenizer is adopted in this work.

The code below is used to split the given documents into a set of constituent individual words.

```
from nltk.tokenize import word_tokenize

for doc in documents:

    words = word_tokenize(doc)
```

Stop word removal: - Stop words are extremely common words across document collections that have no discriminatory power to express documents (Ceri et al., 2013). Articles, prepositions, conjunctions, adverbs and pronouns are some examples of Stop words. Removing stop words reduce the size of indexing file and it also improves the overall efficiency and makes effectiveness in result retrieval (Singh & Saini, 2014). Stop words are language dependent. Therefore, English stop words were imported from NLTK corpus in this work and removed from documents.

Luhn (1958) suggested that both extremely common (upper cut-off) and extremely uncommon (lower cut-off) words were not very useful for document representation & indexing. That means significant words exist between upper cut-off and lower cut-off. So in this study besides ignoring stop words, rare words (words that are exist only once) are also removed from document for indexing.

The following sample code shows the process of removing stop words from documents and identifies words than are not stop words.

```
from nltk.corpus import stopwords

words = []

stopWords = set(stopwords.words('english'))
```

```
for word in documents:
    if word not in stopWords:
        words +=word
```

Stemming- stemming is the process of reducing tokens into their root of words to recognize morphological variation (Ceri et al., 2013). The process involves removal of affixes (i.e. prefixes & suffixes) with the aim of reducing variants to the same stem. Often removes inflectional & derivational morphology of a word.

Inflectional morphology: vary the form of words in order to express grammatical features, such as singular/plural or past/present tense; for example, record →records, record → recording.

Derivational morphology: makes new words from old ones; for example, creation is formed from create, but they are two separate words; and also, destruction → destroy

In general, stemming phase of text operation is used to extract the sub-part of a given word to have exactly matching stems, to minimize storage requirement and maximize the efficiency and effectiveness of filtering Model (Singh & Saini, 2014).

There are different rule based stemmer algorithms. For example, in this work Snowball English stemmer algorithm is used with the reason that it is better than original porter stemmer. For example, stemming word 'generously' with Snow-ball English stemmer brings the word to its root 'generous' while porter stemmer brings to 'gener' which is too expands the word and affects the performance of the system.

The following python source code (see figure 3.3) is sample of Snowball algorithm that removes 'es', 'ed', 'es' and 's' from words ends with 'sses', 'ied', 'ies' and 's' respectively.

```
 __step1a_suffixes = ("sses", "ied", "ies", "us", "ss", "s")

for suffix in self.__step1a_suffixes:
    if word.endswith(suffix):
        if suffix == "sses":
            word = word[:-2]
            r1 = r1[:-2]
            r2 = r2[:-2]
        elif suffix in ("ied", "ies"):
            if len(word[:-len(suffix)]) > 1:
                word = word[:-2]
                r1 = r1[:-2]
                r2 = r2[:-2]
            else:
                word = word[:-1]
                r1 = r1[:-1]
                r2 = r2[:-1]
        elif suffix == "s":
            for letter in word[:-2]:
                if letter in self.__vowels:
                    step1a_vowel_found = True
                    break
            if step1a_vowel_found:
                word = word[:-1]
                r1 = r1[:-1]
                r2 = r2[:-1]
```

Figure 3.4 Sample algorithm of Snowball stemmer

3.5 Indexing

Indexing is carried out after text extraction and text preprocessing. It is a method used to speed up access to desired information from document collection as per users' information need such that it enhances the efficiency in terms of time for recommendation. The most popular data structure employed by IR systems is the inverted file. An inverted file is a data structure for efficiently indexing texts by their words. That means, inverted file is a list of words where each word is followed by the identifier of every text that contains the word. The frequency of each word in a text is also stored in this structure (González, 2008). In this work, python module called "gensim" is used to generate index file automatically (bag-of-words for document representation).

Gensim in its full is "generate similar". Originally it is developed in 2008 as a collection of various Python scripts for the Czech Digital Mathematics Library dml.cz (Rehurek & Sojka, 2010).

"Gensim is a free Python library designed to automatically extract semantic topics from documents, as efficiently (computer-wise) and painlessly (human-wise) as possible.

It is designed to process raw, unstructured digital texts ("plain text"). The algorithms in gensim, such as Latent Semantic Analysis, Latent Dirichlet Allocation and Random Projections discover semantic structure of documents by examining statistical co-occurrence patterns of the words within a corpus of training documents. These algorithms are unsupervised, which means no human input is necessary – you only need a corpus of plain text documents.

Once these statistical patterns are found, any plain text documents can be succinctly expressed in the new, semantic representation and queried for topical similarity against other documents"(Rehurek & Sojka, 2010).

3.6 Vector space model

For identifying documents matching with the users' profile and recommend for users, we used vector space model that apply term weighting and similarity measure.

- **Term Weighting**

After words are indexed, term weighting of words in the document should be calculated to prioritize words according to their importance. Therefore, all term weighting factors described under chapter two in section of Vector space model is applied in this work by adopting gensim module in addition to string-matching model. In gensim, different vector space model algorithms are implemented including latent semantic analysis (LSA) and TFIDF Model (Rehurek & Sojka, 2010).

TFIDF term weight is used in this work. Because, Term-weighting helps to apply best matching that improves quality of recommendation set. Essentially, TF-IDF term weighting works by determining the relative frequency of words in a single document compared to the inverse proportion of that word over the whole corpus. Automatically, this calculation determines how relevant a given term is in a particular document. Words that are common in a single or a small

group of documents tend to have higher TFIDF numbers than common words such as articles and prepositions (Ramos, Eden, & Edu, n.d.).

- **Document similarity with information need**

There are many document similarity measurements to calculate similarity between two documents or between user information and document. The most common similarity measure for the Vector Space Model is cosine similarity, which measures cosine of the angle between two vectors in the vector space (Rehurek, 2011). The similarity implemented in gensim is also cosine similarity since it is a standard similarity measure in Vector Space Modeling and is adopted here. The formula of cosine similarity is given below

$$sim(d_j, q) = \frac{\sum_{i=1}^{n} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \sqrt{\sum_{i=1}^{n} w_{i,q}^2}} \text{.....} \quad \text{3.1}$$

Where w_{ij} is weight of term i in document j and w_{iq} weight of term i in user's information need.

After similarity between information need and document is determined, documents are sorted according to decreasing order (from highest to lowest) of their similarity score to user information need. Finally high ranked documents are recommended for users based on identified threshold (similarity score greater than 0.0884).

CHAPTER FOUR

IMPLEMENTATION OF THE PROTOTYPE SDI SYSTEM

In this work, different matching algorithms are implemented such as string-matching, vector space model (TFIDF and LSA) for developing SDI for academic library of Jimma university. Based on experimentation and evaluation result, the best algorithm was selected for the final SDI system developed.

4.1 Architecture of SDI System

The overall architecture of proposed SDI system is shown in figure 4.1 below. It depicts how the prototype SDI works during recommendation of newly arrived books.

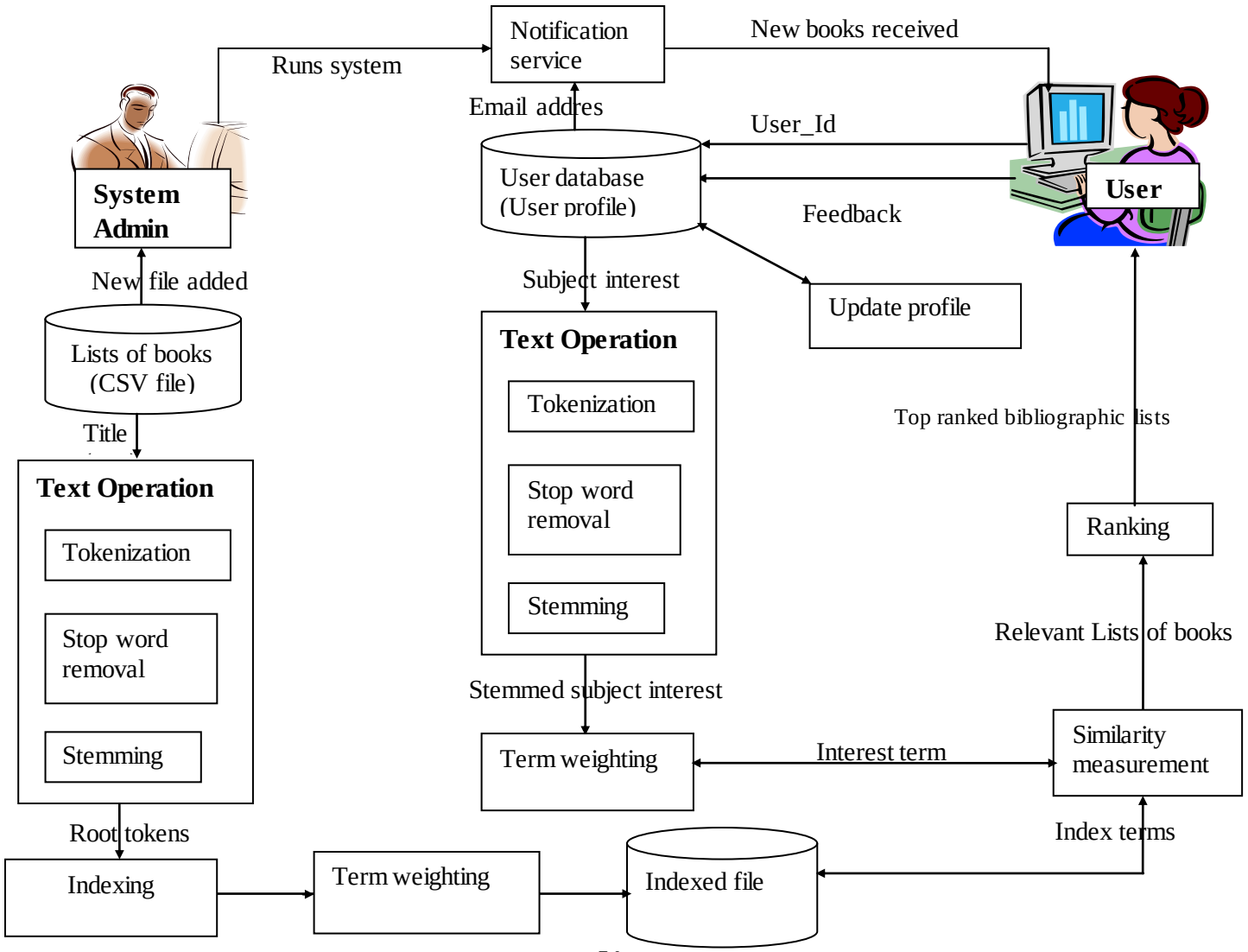


Figure 4.1 Architecture of the system

Once SDI system is designed and service is provided, new users are registered to the system and existing users are insert their user ID to access the SDI service after receiving notification of file change from SDI system administrator. As per the User_Id, the system recommends related bibliography of books for the users based on similarity measurement. As User_Id is entered, prototype of the system selects user information need by User_Id and matches to the documents received by library using similarity measurement. If relevant documents are found within the file, then the prototype rank the relevant retrieved documents on their decreasing order of similarity (top on the first). Next, the prototype recommends relevant document based on threshold determined, which similarity value between user information need and documents is. Then users are deciding whether recommended items are relevant or not. Based on his/her decision the next action is left for user either leaving out the system or updating information need.

4.2 Opening file and selecting Title

To open the file first the researcher defines the file path which CSV files are stored in. then as file is added to defined path or directory to store received books, our system immediately opens added file for input and file which used before is changed to new file automatically. We implemented an age-based priority queue that processes files from oldest to newest based on time from which file is saved to directory. The figure 4.2 shows the algorithm to get oldest and newest file from the directory.

```
def get_oldest_file(oldfile, _invert=False):
    gt = operator.lt if _invert else operator.gt
    if not oldfile:
        return None
    now = time.time()
    oldest = files[0], now - os.path.getctime(oldfile[0])
    for f in files[1:]:
        age = now - os.path.getctime(f)
        if gt(age, oldest[1]):

            oldest = f, age
    return oldest[0]
def get_youngest_file(newfile):
    return get_oldest_file(newfile, _invert=True)
```

Figure 4.2 source code to get oldest and newest file

From algorithm, we have two functions "get_oldest_file" and "get_ youngest_file". We call get_ youngest_file function to open newest file. One thing we have to note here is that, how much long the file should be stay as newest? Let's give conceptual definition to the function name first to understand the period in which the file is newest.

Youngest means nothing is born after; probably from the same mother. Therefore, when we say youngest file, we mean that no any file is saved to the same directory with the same file format after that file. The file may not be necessarily new since the implementation of newest file is age-based priority. For example, file saved to directory before one year is newer than which is saved before one year and one month. User may look at the file at least twice a year and get the same recommendation if there is no new file added after even year elapsed.

The following figure 4.3 shows that implementation in python When CSV file is opened and read, as well as when title attribute is selected and appended to list

```
import csv
lines = []
files = glob.glob('D:\Recieved Books\*.csv')
books = open(get_youngest_file(files),'r')
reader = csv.reader(books)
for row in reader:
    if reader.line_num == 1:
        continue
    my_list = row[1]
    lines.append(my_list)
```

Figure 4.3 python code to open CSV file and selects title.

4.3 Selecting user information need

To connect database with python, we need separate MySQL database module called connector/python. We installed mysql-connector-python-2.1.3-py2.7-win32.msi (package name) by MSI installer (Microsoft/windows installer) after downloading the package/module from url: https://dev.mysql.com/downloads/connector/python/.

Once the package is installed to the directory C:\Anaconda2\Lib\site-packages, we have to import the module to connect the database as shown below.

import mysql.connector **as** sql. Here, we named mysql.connector to sql.

A connection with MySQL server can be established using either the mysql.connector.connect() function or the mysql.connector.MySQLConnection() class (Oracle and/or its affiliates, 2016): in this work sql.connect() function were used since we named mysql.connector to sql to make typing simple.

Table 4. 1 Arguments for Connector/Python (adopted from Oracle and/or its affiliates, 2016)

Argument Name	Default	Description
user (username*)	root	The user name used to authenticate with the MySQL server.
password (passwd*)	text, number, characters	The password to authenticate the user with the MySQL server.
database (db*)	juls	The database name to use when connecting with the MySQL server.
host	127.0.0.1	The host name or IP address of the MySQL server.
port	3306	The TCP/IP port of the MySQL server. Must be an integer.

In this research, we assigned these arguments to variable "config" and call the variable to connect.

```
config = {
  'user': 'root',
  'password': '************',
  'host': '127.0.0.1',
  'database': 'juls',
  'raise_on_warnings': True,
}
db = sql.connect(**config)
```

Figure 4.4 connecting to database

After we connected to database, now we can create table, insert data into table, retrieve data from table, and shortly, we can apply all SQL statements to database. At this level, the researcher used

select statement since table is created and information is inserted into table during profile creation manually using MySQL workbench 6.3.

```
Id = str(raw_input("insert User_Id: ")).lower()

val= cursor.execute("SELECT MajorSubjectInterest, SubsidiarySubjectInterest from user WHERE User_Id = '%s'" % (Id))
result = cursor.fetchall()
for r in result:
    Major = r[0]
    Subsidiary = r[1]
    if Subsidiary =="":
        need = Major
    else:
        need = Major + '; ' + Subsidiary

    needs = need.split(';',7)
    queries.append(needs)
```

Figure 4.5 extracting subject interests from database and splitting into sentences

4.4 Building an Index

Indexing is used to speed up access to desired information from document collection as per users' information need. Building an index from a document collection involves several steps, from gathering and identifying the actual documents to generating the final indexing structures (González, 2008).

As noted by González (2008), there are sequences of steps involved in building an index. The first step is providing indexable items that are coming from sources of information. This is followed by determining the character sequence inside each document (encoding mechanism has to be determined in order for a correct treatment of the text). The third step is deciding the granularity of the index (occurrences of terms in documents and where a term occurs in documents). The fourth step is transforming words from a document to indexable units called tokens (tasks of text operation or preprocessing (linguistic techniques)). Finally, the fifth step in the index building process is creating two data structures called dictionary and postings file.

In this work, each step is implemented in python as follows. First, the text is preprocessed to clean it. Python code presented in Figure 4.6 is used for the text preprocessing

```
def cleanDoc(doc):
    stopWords = set(stopwords.words('english'))
    stemmer = SnowballStemmer('english')
    tokens = WordPunctTokenizer().tokenize(str(doc))
    clean = [token.lower() for token in tokens if token.lower() not in stopWords and len(token) > 2 and token.isalpha()]
    final = [stemmer.stem(word) for word in clean]
    return final
```

Figure 4.6 a python code for text operation

Then we extracted preprocessed texts from title, just we named it document in the code by calling the cleanDoc function. We also removed very rare words from document that appears only once in the whole document.

```
texts = [[word for word in cleanDoc(document)] for document in lines]# lines is lists of title of the books
frequency = defaultdict(int)
for text in texts:
    for token in cleanDoc(text):
        frequency[token] += 1
tokenized = [[tokens for tokens in text if frequency[tokens] > 1] for text in texts]
```

Figure 4.7 Identifying and preprocessing indexable items

After identifying and preprocessing indexable items, we build dictionary which is a lists of unique terms from which documents were constructed by applying approach used by Řehůřek & Sojka (2010) called bag-of-words. Bag-of-words is document representation used to convert documents to vectors. In this representation, each document is represented by one vector where each vector element represents how many times the word appears in the document, Frequency of word.

After we created dictionary, the index file is created which contains dictionary along with information about dictionary terms such as frequency of terms, documents in which the term is found (doc ID) and location.

A fragment of python code written in the implementation of the index file is depicted in figure 4.8 below.

```
dictionary = corpora.Dictionary(tokenized)
corpus = [dictionary.doc2bow(text) for text in tokenized]
corpora.MmCorpus.serialize('D:/Index/newbook.mm', corpus)
corpus = corpora.MmCorpus('D:/Index/newbook.mm')
tfidf = TfidfModel(corpus)
index = similarities.MatrixSimilarity(tfidf[corpus])
```

Figure 4.8 index file construction

4.5 Matching and recommending

Now we have major components of SDI service; Items (documents) and user information need (user subject interest) and indexed them in the situation they are favorable for final SDI service. But to provide final SDI service, matching should be done between two components to recommend relevant document or item to users (targeted one).

We implemented different filtering models to perform the action of matching and recommending.

String-matching model

On the first step we implemented string-matching model for identifying documents matching with user interest.

A document would match the information need of a user if at least one user-specified string exists in the document. So the method is search for document as per the term is found in the document. That means for every term in the query, it searches the term in the document one by one and if term is exist, document is printed. The code below shows string-matching algorithm

```
for s in cleanDoc(str(query).split(" ")):
    for doc in lines:
        if s in cleanDoc(doc.split(" "))
            print all_docs[lines.index(doc)], '\n'
```

Figure 4.9 string-matching algorithm

However, string-matching model is unable to sort documents and so it is difficult to judge relevant document. The detail is explained in section 5. To overcome the problem, we implemented vector space model.

Vector Space Model

As stated under section For identifying documents matching with the users' profile and recommend for users, we used vector space model that apply term weighting and similarity measure.

Term Weighting, in gensim several VSM approaches to representing documents as vectors were tried by (Řehůřek & Sojka, 2010) like LSA model and TFIDF model and we adopted the same models to develop our SDI system.

In this work, we adopted the model as follows.

TFIDF Model as mentioned earlier, is document-term sparse matrix term weighting. From knowledge of the researcher, it is widely used Vector space model term weighting algorithm in the field of information filtering and information retrieval. Also in this work, the model is selected as best one with the reason that will be discussed latter based on the experiment.

```
from gensim.models.tfidfmodel import TfidfModel
tfidf = TfidfModel(corpus)

for query in queries:
    new_vec = dictionary.doc2bow(cleanDoc(query))
    vec_tfidf = tfidf[new_vec]
index = similarities.MatrixSimilarity(tfidf[corpus])
similar = index[tfidf[vec_tfidf]]
sims = sorted(enumerate(similar), key=lambda item: -item[1])
count =0
for (id, score) in sims:
    if score > 0.0884:
        count +=1
        print count
        print "Title: ", str(lines[id]), '\n'
        print "Author: ", str(Author[id]), '\n'
        print "Edition: ", str(Edition[id]), '\n'
        print "ISBN: ", str(ISBN[id]), '\n'
        print "Publication Date: ", str(Pub_Date[id]), '\n'
        print "Publisher: ", str(Publisher[id]), '\n'
        print "Binding: ", str(Binding[id]), '\n'
        print "Quantity: ", str(Qty[id]), '\n'
        print "-----------------------------------------------------------------------------------------------"
```

Figure 4.10 how TFIDF model do matching and print related documents

LSA models term co-occurrence model. It assumes terms that co-occurred in document are semantically related

```
from gensim.models.lsifmodel import LsiModel
tfidf = LsiModel(corpus, id2word=dictionary, num_topics=2)

for query in queries:
    new_vec = dictionary.doc2bow(cleanDoc(query))
    vec_lsi = lsi[new_vec]
index = similarities.MatrixSimilarity(lsi[corpus])
similar = index[lsi[vec_lsi]]
sims = sorted(enumerate(similar), key=lambda item: -item[1])
count =0
for (id, score) in sims:
    if score > 0.0884:
        count +=1
        print count
        print "Title: ", str(lines[id]), '\n'
        print "Author: ", str(Author[id]), '\n'
        print "Edition: ", str(Edition[id]), '\n'
        print "ISBN: ", str(ISBN[id]), '\n'
        print "Publication Date: ", str(Pub_Date[id]), '\n'
        print "Publisher: ", str(Publisher[id]), '\n'
        print "Binding: ", str(Binding[id]), '\n'
        print "Quantity: ", str(Qty[id]), '\n'
        print "-----------------------------------------------------------------------------------------------------"
```

Figure 4.11 how LSA model do similarity and print related documents

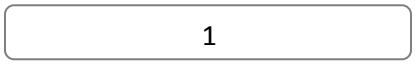
4.6 Feedback and profile updating

After necessary information is selected and supplied to the clients, there should be a mechanism to receive a feedback on usefulness of supplied information. This mechanism is asking users to fill out feedback questionnaires in order to find out usefulness of packages and update user's profile (Hossain & Islam, 2008).

We provided one question with three options that asks whether user is satisfied with SDI service and want to change his/her user's information need. First we put the options 1-3 and next we asked the user which he/she is likely to choose as follow. The user selects only option number.

1) I'm satisfied and want to stay with my subject interest
2) I'm satisfied but want to change my subject interest
3) I'm not satisfied and want to change my subject interest

Which one of above statements describes you? Choose number



Here, the user selects 1st option. That means, he/she didn't want to change his/her information need. So the system accepts his/her opinion and registered to database that he/she is satisfied with the system without next action except acknowledging user for his/her opinion.

But if user selects 2 or 3, the system allows user to change his/her major subject interest and subsidiary subject interest.

After the system is rerun, the system recommends documents related to new user's information need.

For example, see the following figures 4.12a and 4.12b before profile is updated and after profile is updated for user ID juls/06. In the beginning, the user user's information needs were: Information retrieval system, information filtering and semantic web technology. So figure 4.12a illustrates some of related documents to these user's information needs.

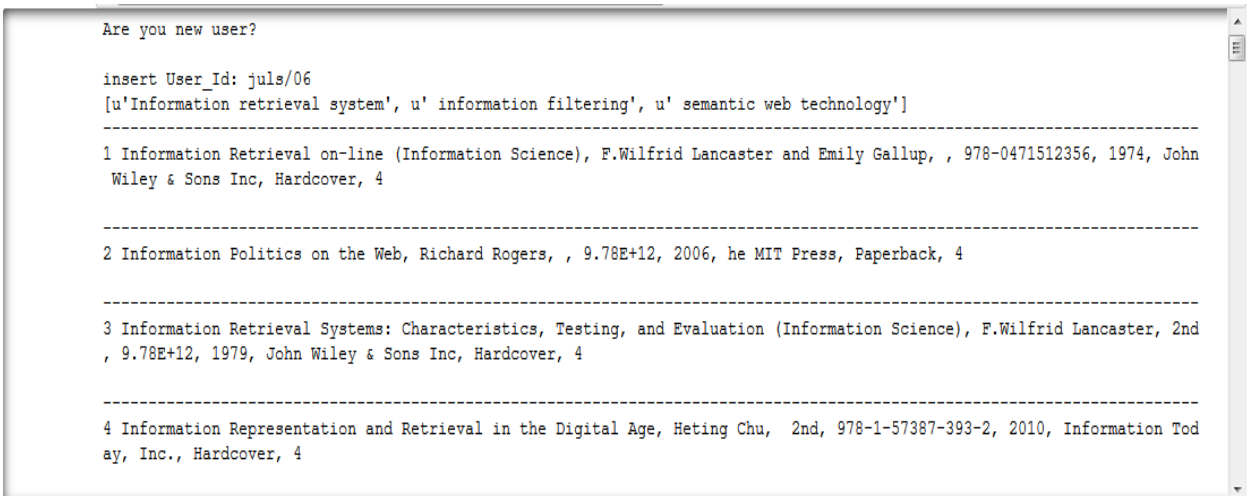


Figure 4.12a result before profile is updated

Now the user selects 2 like this 2 and updated his user's information need to knowledge management (MajorSubjectInterest) and knowledge acquisition process (SubsidiarySubjectInterest). See some related document recommended to these new subject interests in figure 4.12b

```
Are you new user?
    n
insert User_Id: juls/06
[u'knowledge management', u' knowledge acquisition process']
--------------------------------------------------------------------------------------------------------------------------
1 The Enterprise Of Knowledge , Issaac Levi , , 9.78E+12, 1983, The MIT Press, Paperback, 2

--------------------------------------------------------------------------------------------------------------------------
2 Knowledge Management: An Introduction, Kevin C Dezoua, , 9.78E+12, 2011, facet Publishing, Paperback, 4

--------------------------------------------------------------------------------------------------------------------------
3 The Knowledge Management Toolkit: Orchestrating IT, Strategy, and Knowledge Platforms , Amrit Tiwana, 2nd, 978-01300922
43, 2002, Prentice Hall, Hardcover, 4

--------------------------------------------------------------------------------------------------------------------------
4 Knowledge Management in Theory and Practice, Kimiz Dalkir, 2nd, 978-0262015080, 2011, The MIT Press; second edition , H
ardcover, 4
```

Figure 4.12b result after profile is updated

This function of receiving feedback and updating profile is programmed in python with fragment codes illustrated by figure 4.13 below.

```
db = sql.connect(**config)
cursor = db.cursor()
print "\n============Express your openion=============\n"
print "1) I'm satisfied and want to stay with my subject interest"
print "2) I'm satisfied but want to change my subject interest"
print "3) I'm not satisfied and want to change my subject interest"
openion = raw_input("\nWhich one of above statents describes you? choose number \n ")
Identity = Id#raw_input("insert your ID again")
cursor.execute("insert into openion(User_Id, openions) values('%s','%s')" %(Id, openion))
db.commit()
if openion == '2' or openion=='3':
    MajorSubject = raw_input("insert your major subject interest:\n ")
    SubsidiarySubject= raw_input("insert your subsidiary subject interest:\n ")
    cursor.execute("update user set MajorSubjectInterest ='%s' where User_Id = '%s' " %(MajorSubject,Identity))
    cursor.execute("update user set SubsidiarySubjectInterest = '%s' where User_Id ='%s' " %(SubsidiarySubject, Identity))
    db.commit()
```

Figure 4.13 a python code to receive feedback and update profile

4.7 Notification or alerting service

Because the SDI systems are most interested in new content arriving at an information provider, knowledge of the event of new content arrival is of key importance (O'Neil, 2001). Thus our system is attempted to detect new file arrived at library immediately and consider the file for further recommendation. However, this is known if and only if users were gone to the system

and use the SDI system. That means if the documents are arrived to the library let's say twice before users used the SDI system, user lost information about the first documents arrived to library since the implementation of newest file is age-based priority and therefore, the first file is older than the second one.

Therefore, the researcher implemented another separate administrative system that can notify users at arrival of document into library. The system need librarian to operate, just running the system during this prototype or one click (notify user) if it is implemented through user interface.

There are two general solutions to implementation of conveying this event to interested user. First, the system sends list of relevant documents subscribed users in the similar way SDI system recommends users. However, the format in which documents were sent to users is not favorable to read recommended document since it sends in plain text. So the researcher leaves this solution to the next researcher to improve. Second, the system transmits a simple message that redirects users to address of SDI system and denoting content change to entities that have "subscribed" to receive such messages. Subscribed users are those users filled their email address during profile creation. It is up to users to fill up their email address. If the user has no email address, no problem with the system, it is implemented in a manner that it can sends message to only users those have email address and skips users those have no email address.

```
import smtplib
from email.MIMEMultipart import MIMEMultipart
from email.MIMEText import MIMEText
maillist =[]
toaddr = cursor.execute("SELECT e_mail from user")
result = cursor.fetchall()
for r in result:
    email =r[0]
    maillist.append(email)
fromaddr = "juls.acquisition@gmail.com"
toaddress = [address for address in maillist if address is not None]# to only users those have email address
toadd = ', '.join(toaddress)
msg = MIMEMultipart()
msg['From'] = fromaddr
msg['To'] = toadd
msg['Subject'] = "New Arrivals"

body = "Dear Library user, there is new books arrived to the library. go to address to SDI service to get your specific need


msg.attach(MIMEText(body, 'plain')),'\n'

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(fromaddr, "julsacquisition")
text = msg.as_string()
server.sendmail(fromaddr, toadd, text)
server.quit()
```

Figure 4.14 how notification service is implemented in python

CHAPTER FIVE

EXPERIMENTATION AND EVALUATION RESULT

In this research, the data set contains lists of bibliographic information from different domains, such as Accounting, Biology, Law, Pharmacy, Civil engineering, Veterinary medicine, Sociology, Psychology, etc. It was divided into training data set and testing data set in the ratio of eighty percent (80%) by twenty percent (20%), respectively.

5.1 Subject interest selection

As we mentioned under section 3.3, user subject interest is ranged from one (1) to eight (8). Therefore, by considering the representativeness of whole users information need, the researcher selected shortest subject interest (query), medium subject interest (query), and longest (eight statements) subject interest (query) purposely for experimentation. One information need which is selected from file and registered with user Id JU/PHMS/28 to database was also identified specially to experiment whether ranking of documents were done properly or not since the document from which that information need obtained was expected to be on the top (on the first) during recommendation.

In order to make experiment, subject interests of ten users were selected. Recommended documents are marked across each subject interest as either relevant or irrelevant to have relevance judgments. The term relevance judgments indicate users' decision on whether a document satisfies their information needs of a specific subject interest. So, those users with identified subject interests were involved in determining the relevance of documents recommended as related documents for subject interests identified. The main importance of having identified user subject interest is to evaluate the performance of the system.

Table 5.1 test/experimental subject interests

User ID.	Subject interest	Description
JU/PHMS/10	microbiology	Shortest
JU/IT/03	Instrumentation	Shortest
JU/CNS/15	Digitization and digital libraries	Short
JU/CAVM/03	Alternative processing technologies to improve quality of foods; Application of processing methods to modify functional properties of foods; improving traditional processing methods for better food safety and enhanced quality	Medium
JU/IT/12	Modeling of land surface processes and interactions with the atmosphere; Earth Observation of water cycle and applications in climate and ecosystem and water resources studies; Developing and applying hydrologic modeling approaches for Water Resources Management; Hydro climatology (floods and droughts); Geo-information Science & Remote sensing applications in hydrology, and climate change	Longest
JU/CSSH/02	Sociolinguistics, Multilingualism, Multilingual Education, Psycholinguistics, Language psychology; Translation; Communication Theory; Public Relations, Journalism; Discourse Analysis; Four Language Skills (Listening, Speaking, Writing and Reading); Pragmatics; Human development and Attitude.	Longest
JU/CLG/01	Environment and development; Sustainable tourism and development; Gender and development; Environmental governance, politics, conflict dynamics and development	Long
JU/PHMS/28	Drug testing in criminal justice	From document
JU/BECO/03	Entrepreneurship; Organizational culture; Leadership;	long

	Marketing Management; Human Resource Management; Operations Management	
JU/CNS/04	Drug resistance in Plasmodium vivax; Impact of immunotoxic pollutants on infectious disease; Anti-plasmodial activities of medicinal plants	medium

By using these identified subject interests, the researcher made experiment on each models (string model, TFIDF of VSM model, and LSA of VSM model) implemented and found that result is different for each models.

5.2 System performance evaluation

System performance is evaluated in terms of effectiveness and efficiency. By effectiveness, it means the level up to which the given system attained its objectives. Thus in SDI system, effectiveness may be measure of how far it can recommend relevant information while withholding non-relevant information. On other hand, Efficiency means how economically the system is achieving its objectives. In an SDI system, efficiency can be measured by factors such as cost. The cost factors are to be calculated indirectly. They include factor such as response time, time taken by the system to provide an answer. User effort, the amount of time and effort needed by a user to interact with the system and analyzed the output retrieved in order to get the correct information and storage needed to provide the system.

For measuring the performance of recommender algorithms, measures originating from statistics, machine learning and information retrieval are used (Ababor, 2003b). Most of the time, effectiveness of the system is measured in terms of recall, precision and F-measure while efficiency is measured by user.

So we measured effectiveness of our SDI system by using recall, precision and F-measure.

Recall can be defined as the fraction of relevant items that are also part of the set of recommended items (Isinkaye, Folajimi, & Ojokoh, 2015) and computed as

$$\text{Recall} = \frac{Correctly\ recommended\ items}{Total\ useful\ items} \quad \text{----------5.1}$$

Precision is the fraction of recommended items that is actually relevant to the user and computed as

$$\text{Precision} = \frac{Correctly\ recommended\ items}{Total\ number\ of\ recommended\ items} \text{ ----------5.2}$$

F-measure helps to simplify precision and recall into a single metric. The resulting value makes comparison between algorithms and across data sets very simple and straightforward (Isinkaye et al., 2015).

$$\text{F-measure} = \frac{2PR}{P+R} \text{ ----------5.3}$$

Where p is precision and R is recall

The evaluation process is gone by taking relevance judgments and calculating recall, precision and F-measure of each top ten recommended documents for each subject interest or users' information need identified in table 5.1. Then we took the average value for each subject interest as presented in table 5.2 and table 5.3, under each model.

In this study, different technique such as string-matching and TFIDF model are evaluated to develop SDI system. However, evaluation of LSA model shows that no relevant document ranked from one to ten. That means zero recall and zero precision is registered., this is because as stated in the literature (Rehurek, 2011), the idea behind LSA is term co-occurrence. Remember again, the objective of this study is to recommend bibliographic information of the book to users rather than full contents of the book. Therefore, terms co-occurrence is rare in such short term documents. As described earlier, we used title of the book to compare similarity with users' information need. Through our observation in this study, the maximum numbers of words found in one title are eight without stop words.

So we realized that deficiency of LSA model identified by (Rehurek, 2011; Rosario, 2000) listed as folllows, through our expriment.

- Choosing the optimal value for the k parameter (the latent space dimensionality) is not obvious. In IR practice, the value is typically set to several hundred, but depends on the application as well as the structure of the input corpus

- Topics are not interpretable. In other words, by looking at the m-dimensional vector of a particular topic t_i, $i \leq k$, it can be hard to assign a human label to the theme connecting the highest scoring terms

- Using LSI vectors, we can no longer take advantage of the fact that each term occurs in a limited number of documents (the main factor in our case), which accounts for the sparse nature of the term by document matrix.

- With LSI, the query must be compared to every document in the collection. So the efficiency is slow down.

- **Experiment and evaluation result on string-matching model**

String-matching model assigns equal weights to all terms in the query (Renganathan, Ajit & Suptendra, 2013). From experimental result, we realized that it searches for documents to terms in the subject interest sequentially by order of terms in the subject interest or information need one by one. That means, documents that have first term of subject interest are searched first, document that have second term of subject interest are searched next, after searching for document containing first term is finished, and continue searching for documents containing the end of subject interest term on the end. As a result, document that has more than one term of subject interest printed out times of subject interest terms it contains; if it contains two terms of subject interest it is displayed two times, if three, three times.

So let us show the result with simple example just by taking one short subject interest identified. We selected subject interest of user Id. JU/PHMS/28 which is "Drug testing in criminal justice" since it is selected from one document. So document containing this subject interest is expected to be displayed as recommended book four times (here three times among ten document) since, subject interest contains four terms without "in", which is stop word.

The result displayed is illustrated in the following figure 5.1

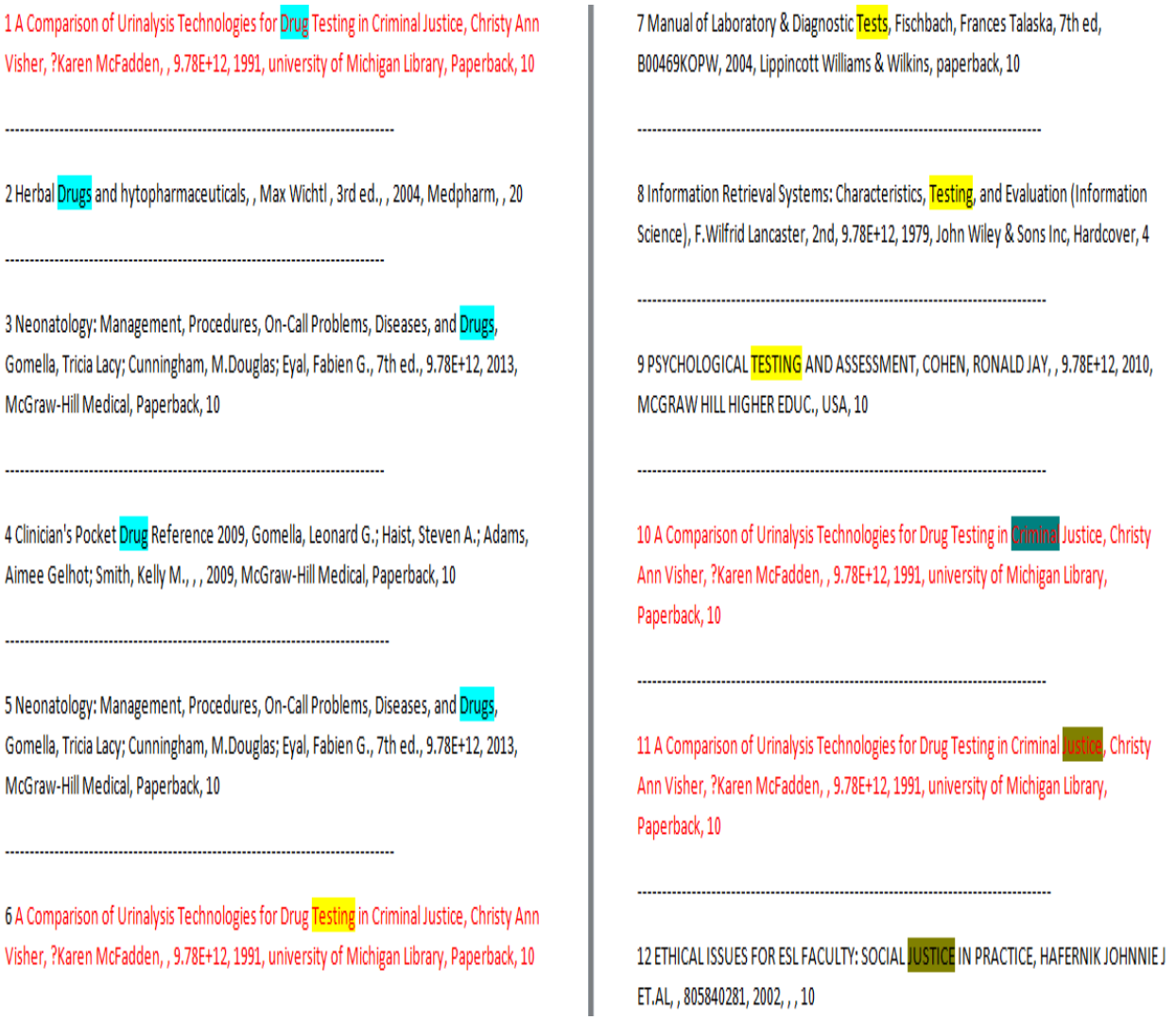


Figure 5.1 recommended document for user by string-matching model

From figure 5.1, we colored duplicated document with red and highlighted terms with different colors. As it is observable, documents are recommended in the sequence of terms they contain from subject interest. Therefore, there are five (5) documents containing term "Drug", four (4) documents containing term "Testing", one (1) document containing term "Criminal" and two (2) document containing term "Justice". Remember that terms are stemmed.

Figure 5.1 show that string-matching model is unable to sort documents rather it searches for document in sequence for which the term is arranged in the user information need. In fact, certain terms have little or no discriminating power in determining relevance (Manning, Raghavan, & Schütze, 2009). The main aim of this study is recommending bibliographic lists of

books; not full content and string-matching algorithm attempts to find the similarity between user information need and Title. Therefore, the sequence in which terms are arranged in user information need is not promising in recommending relevant documents. Because there are many words or terms used in different disciplines. Let us take some typical examples: words such as "FUNDAMENTAL" and "PRINCIPLE" are some frequently terms used in all domains specially for titling documents. For example, in our testing document used in this study, there are thirty two (32) documents containing word "fundamental". Some of these documents from different domain are:

- Fundamental of collection development and Management
- Fundamentals of Database Systems
- Fundamentals of Management
- Fundamentals of Food Process Engineering
- Conducting Polymers, Fundamentals and Application and so on

By only putting this word on the beginning of users' information need, user is recommending these thirty two (32) documents on the first without considering the importance of terms in the document. So experimentally we realized that string-matching model is worries model since all terms are treated equally and sorting of documents are not possible. Also documents are recommended redundantly.

Summary of experimental result of SDI based on string-matching model for top ten recommended documents is presented in table 5.2 below. System Performance of String-matching model used in developing SDI is also presented in terms of Precision, Recall and F-measure.

Table 5.2 Experimental and evaluation result using string-matching model

User ID.	Retrieved	relevant	Recall	Precision	F-measure
JU/PHMS/10	10	10	1	1	1
JU/IT/03	9	8	0.89	0.8891	0.8895
JU/CNS/15	10	9	0.9	0.86	0.88
JU/CAVM/03	10	4	0.4	0.41	0.40
JU/IT/12	10	0	0	0	0
JU/CSSH/02	10	1	0.1	0.29	0.15
JU/CLG/01	10	2	0.2	0.256	0.2245
JU/PHMS/28	10	5	0.5	0.79	0.6124
JU/BECO/03	10	7	0.7	0.8845	0.7815
JU/CNS/04	10	5	0.5	0.51	0.5049
Average			**0.519**	**0.5889**	**0.5517**

From table 5.2, we can understand the performance of the system is very poor for longest subject interest. However, for shortest subject interest, the system is somewhat encouraging. Since longest subject interests contain more words and string-matching model cannot sort documents, performance evaluation done on top ten recommended items indicates that string-matching model is poor. This means in string matching modeling we cannot determine where relevant documents were found, is it on the top, medium or bottom? In practice, it is better if most relevant documents should be at the top to make a user selects relevant documents easily.

In general, as shown in table 5.2, string-matching model registered 52% recall, 58.89% precision and 55.17% F-measure on average.

The result indicates string-matching model is less in finding and recommending relevant documents since both recall and precision registered were lower. However, among recommended documents 58.89% of them are recommended successfully which is greater than recall. In all cases, the result shows that greater precision were registered than recall. To get detail of evaluation result along with experimental result, refer appendix III under appendices

- **Experiment and Evaluation result on VSM using TFIDF term weighting**

The application of Vector space model algorithm or TFIDF weighting model is also evaluated by using the same subject interests used in string-matching model on top ten recommended documents. Table 5.3 below shows the relevance and performance of the SDI system by calculating recall, precision and F-measure.

Table 5.3 performance Measurement of TFIDF term weighting of VSM using Precision, Recall and F-measure

User ID.	Relevant	Retrieved	Recall	Precision	F-measure
JU/PHMS/10	10	10	1	1	1
JU/IT/03	9	8	0.89	0.8891	0.8895
JU/CNS/15	10	9	0.9	0.8902	0.895
JU/CAVM/03	10	6	0.6	0.616	0.6078
JU/IT/12	10	3	0.3	0.6285	0.4061
JU/CSSH/02	10	8	0.8	0.809	0.8044
JU/CLG/01	10	4	0.4	0.5555	0.465
JU/PHMS/28	9	6	0.67	0.70	0.6846
JU/BECO/03	10	9	0.9	0.9522	0.9254
JU/CNS/04	10	6	0.6	0.836	0.6986
	Average		**0.706**	**0.7876**	**0.7376**

Values under retrieved column are less or equal to the values of the same column of string-matching model. For single term information need, all results are equal with string-matching model. Unlike string-matching model, TFIDF model of vector space model able to sort documents in decreasing or increasing order of their similarity values. Since we can sort documents, we can rank most important documents at the top. After we ranked documents, we can set threshold value on which the system stop recommending document either by total number of document should be recommended (top N document) or by similarity values (Manning, Raghavan, & Schutze, 2009). In this work, we used cosine similarity value greater

than 0.0884 of documents to recommend them. We take this value after we observed and experimented on longest user information need. By our experiment during relevant judgment, we realized that documents recommended after this similarity value are not that much relevant and we cut off from the result.

```
for (id, score) in sims:
    if score > 0.0884:
        related = all_docs[id]
        print related,'\n'
```

Figure 5.2 threshold value

In contrast to string-matching model, in TFIDF model of vector space model, we are not bothering about terms that are frequently found in different documents such as "Fundamental and Principle". Because in vector space model, terms that are found in many documents get lower weight than terms that are found in few documents. In other word, IDF of a rare term is high, whereas the IDF of a frequent term is likely to be low (Manning, Raghavan, & Schutze, 2009). Therefore, we found documents containing those frequent terms at the bottom of our recommended documents and we can judge relevance easily. Nevertheless, it depends on other terms in the users' information need since similarity between users' information need and document is based on total term weight of document. That means as many of terms in the users' information need is found in the document, the document is become more similar to users' information need.

To illustrate the result with example, let us take the same user information need (Drug testing in criminal justice) we used to test string-matching model.

The document contains this information need is expected to be recommended on the first. That means similarity of the document with information need is highest (0.81351322) since the weight (TFIDF) of the document from which the information need accepted is high.

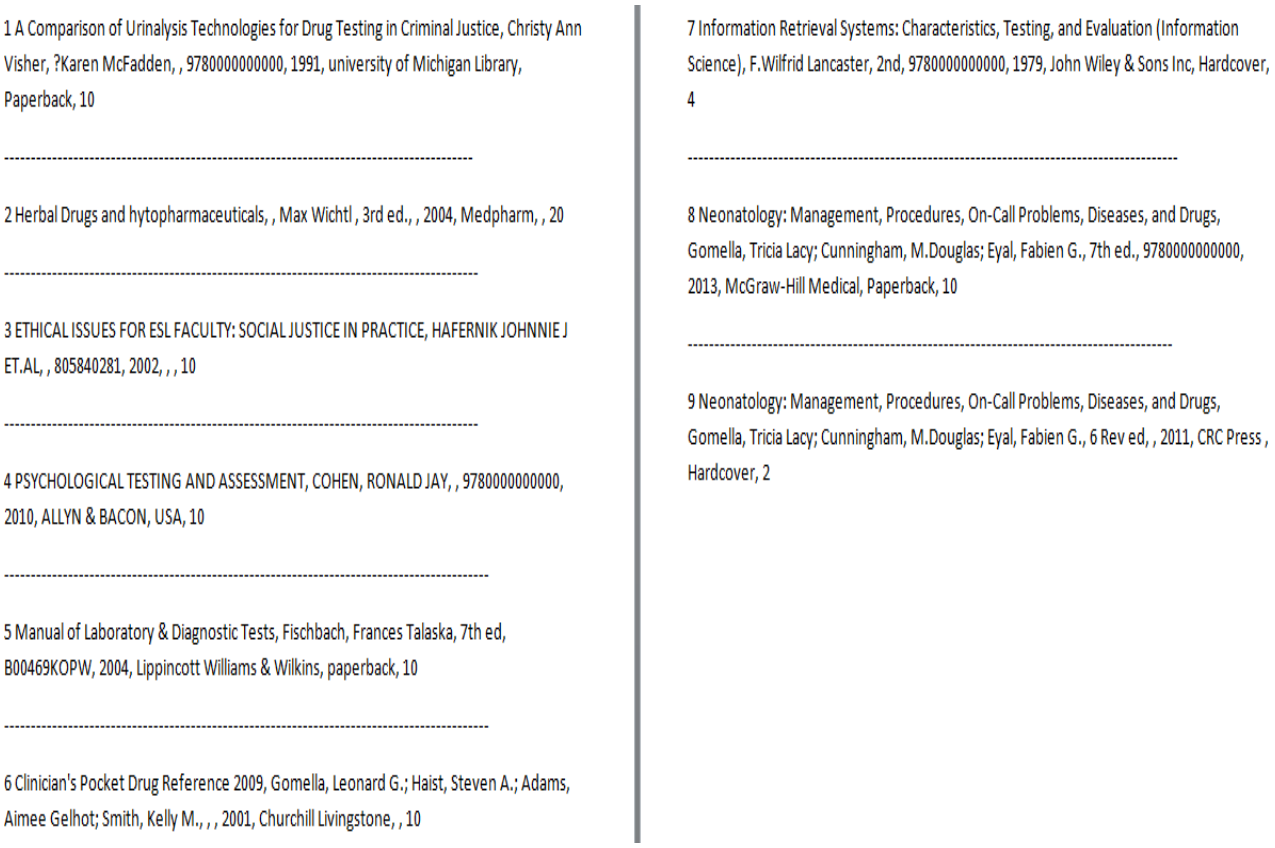


Figure 5.3 recommended document for user by TFIDF of VSM model

As figure 5.3 shows, we have nine (9) documents recommended since three of redundant documents recommended under string-matching model were removed.

From Table 5.3 above, the average results are 70.6%, 78.76% and 73.76% for recall, precision and F-measure respectively.

This indicates that the system is very promising result. Achieving performance of greater than 70% in such dynamic users' information need is interesting result. Especially, achieving precision about 78.76% is good news since among recommended results, 78.76% of them were accurate.

Due to its ability to sort documents based on their similarity measurement, TFIDF term weighting model registered greater performance than string-matching model. That means users found most relevant documents on the top and the order in which documents are ranked has impact on the performance of the system specifically precision. For example, if we have relevant

document recommended on the first, we have recommended 100% accurately to that specific item. That means, from one document recommended we have one relevant document (1/1 =1).

Generally, less recall is registered when compared with precision. Because recall is based on dataset in our case, only title attribute of the dataset. For example, from 977 test dataset used in this research, 493 unique words were indexed. These words may increase if we indexed from whole contents (full text) of the books. Therefore, number of documents recommended depends on content of the document. That is why we appreciated the result found since we have no content from our dataset. For more detail about experimental and evaluation result, see appendix IV under appendices

5.3 User Acceptance Evaluation

We conducted user acceptance evaluation into two different viewpoints; managerial oriented evaluation and user-oriented evaluation.

Therefore, performance evaluation is carried out with Jimma University library users and library managers. To check validity of the system in the library, three of library managers who are also academic staff were selected for evaluation of the system; library director, deputy librarian and library ICT team leader.

Therefore, the researcher provided two types of questionnaire one for managers and other for users. The questionnaire was Likert scale type of questionnaire, which requests respondent the degree of agreements to each evaluation Parameters and users were allowed to tick their option.

Therefore, there are five options for these closed ended questions; strongly disagree, disagree, neutral, agree and strongly agree. Numerically we assigned 1-5 values respectively to calculate average result out of five.

Validity testing with library managers

Table 5.4 validity test by library managers

No	Evaluation Parameters	Performance Value					Average	Decision
		Strongly disagree	Disagreed	Neutral	Agree	Strongly agree		
1	Proposed system can solve the problem at hand in the library				1	2	4.67	Strongly agree
2	The system is easy to use					3	5	Agree
3	The system is efficient in time and memory					3	5	Strongly agree
4	The system is significant in the library					3	5	Strongly agree
						Total average	4.92	S. agree

From table 5.4, 67% of respondents were strongly agreed that the system is solving the existing problem of finding new information while the rest 33% is agreed. In other way, 100% of respondents rated strongly agree that SDI system proposed is easy to use. 100% of the respondents were strongly agreed on efficiency of the prototype SDI system. Finally, all managers (100%) were strongly agreed as proposed system is significant in the library.

In general, based on the evaluation of all the respondents the average performance of the prototype is 98.35%, which can be decided as strongly agree. This performance result from library managers shows that the prototype is valid and can be applied in the library to recommend new arrivals of books for library users.

To evaluate the prototype system with user, we have selected the same ten users those we have used their user's information need in the experimentation part.

User acceptance evaluation

Table 5.5 Users acceptance testing

No	Evaluation Parameters	Performance Value					Average	Decision
		strongly disagree	disagree	neutral	agree	strongly agree		
1	The prototype system presented all relevant items for me				4	6	4.6	Strongly agree
2	I found all relevant items to my subject interest on the top				2	8	4.8	strongly agree
3	I recommended documents as soon as I insert my User_Id					10	5	Strongly agree
4	No more effort is required from me while I use the system to obtain relevant document to my subject interest				3	7	4.7	Strongly agree
5	Recommended items are presented in easy way that I can read and understand				6	4	4.4	Agree
6	Application of such system is very necessary in the library					10	5	Strongly agree
						Total average	4.75	S. agree

From Table 5.5, 60% of users responded strongly agree that prototype system presented all relevant items for them while 40% of them were responded agree. 20% and 80% of respondents rated agree and strongly agree respectively that they get all relevant documents on the top. In the case of time, 100% of user are rated strongly agree as they recommended items as soon as they insert their User_Id. In the case of effort, 30% by 70% selects agree and strongly agree respectively as no more effort is needed to use the SDI system. 60% of respondent agreed up on presentation and readability of recommended documents while remaining 40% strongly agreed. The final evaluation parameter that deals about necessity of applicability of the system to library is evaluated. Accordingly, 100% of respondents were strongly agreed that the application of the system in the library is important.

Generally, based on the evaluation of all users, 4.75 (95%) average performance of the system is registered by users which indicates level of agreement of the users as strongly agree. Therefore, we can decide significance of SDI system has achieved advisable result from users.

5.4 Discussion

As it is described in the evaluation part of this report, encouraging result was achieved in this research/study. Even though the result of string-matching model and TFIDF model of VSM model is equal for single word information need, on average, the highest system performance is achieved by SDI system, which is developed using VSM model with TFIDF weighting. Total precision registered by this approach is 78.76% while recall is 70.6%. This indicates that relevant documents recommended were more usable since better result is registered for precision than recall. The performance of the system is affected by different factors. These factors are the manner user set his/her subject interest and size of collection from which document is recommended. For example, user's information need used in this work is not more convenient to the design SDI system. Because information need of the user is not on specific subject area, rather it is specific to say, title of an article. However, Scope of the core subject interest should be broader subject of the study to scan the core subject matter (Hossain & Islam, 2008). That is the reason why longest subject interest regestered lower result. Therefore, there is a hope that these results can be increased if users are orinted how to set their information need to use SDI service. If collection is large, the performance is increase as observed in this work. That means,

there is a probablity of relevant document to be recommended from large collection. That is why obtained result in this work is appriciated from such limited source of documents.

After best performed system is selected, the system is also tested both by library managers and users. The system is validated by library managers performing 98% of level of agreement; which is highest level of agreement, strongly agree. User's acceptance testing also registered highest level of agreement to evaluation parameters about 95% that can be decided as strongly agree.

Even though users are strongly agreed with evaluation parameters, they wish and suggested new arrival recommendation on their email address. Though the proposed SDI system could tell the delivery of documents on their mail and redirect them to use SDI system, the proposed Prototype SDI system could not attach quality information on their mail. Because of time limitation, the researcher could not cover this issue and therefore, needs further investigation. New document containing bibliographic lists of new arrived books is detected automatically to be used as a source of data to recommend documents. Therefore, users have to check developed SDI system day to day because, if new file is added before user recommended previous document, they loss information about the first documents. Even though this is not major problem in JULS since delivery of books is not frequent, due to designed SDI service can be applied in any academic library, the researcher designed notification service on their email, which tells them the delivery of new file and redirect them to SDI service in parallel with designed SDI service.

Both system evaluation and users evaluation of the prototype indicates that SDI system provides high percentage of relevant documents for users. However, the system is only limited to recommend documents written in English. This is due to the reason that users provide their information need in English and the time is matter to incorporate the application of multi-language in the design of SDI system. Therefore, further research is needed to cover this gap.

Current work is able to registered better performance because of filtering approach we used during data acquisition. Former works such as Ababor (2003) and Porcel, C. et al (2012) are based on rating of items in which users are expected to rate more items as much as possible while our system did not need rating of item except it receives user information need separately and disseminates items based on knowledge of user information need. Our initial aim is to reduce user effort in providing their information need to obtain relevant documents to their information

need. We achieved this aim by allowing users to leave their information need in to library's database and library recommends relevant items to their information need by help of proposed SDI system. Therefore, proposed SDI system is simple to use just like search engine since user is only expected to type their user Id. Hence, the SDI system designed can be called SDI engine.

In this work, we could also able to tackle the problem of data sparse which has impact on the performance of SDI service by providing mechanism which automatically detects new file added to library's storage for the input of our SDI system.

Even though some improvements are required for real implementation of the system in the future, the research objective was achieved in proper manner as expected. The main objective of the study is to propose SDI service after development of SDI system by applying an information filtering approach to make recommendation of newly arrived books for Jimma University Library System (JULS) users. Just it is achieved.

Generally, performance achieved is a witness that this study can overcome the problem of finding latest information in the Jimma University library System (JULS) for users.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The problem of the current information age is not lack of information since dozens of thousands of information is published on the web, as well as acquired by libraries day to day. But identifying relevant information among these bulks of information at a time of release (just on time) is the main concern of the era. Therefore, the application of information filtering in the library and web is attracts many researchers especially in the SDI system.

This study tried to explore the problem of finding new information on time and propose prototype SDI system as a solution by applying information filtering approach. Selective dissemination of information (SDI) is a mechanism of distributing information to concerned user based on his/her information need from user profile.

The study was conducted having main objective of designing an SDI service to make recommendation of newly arrived books for Jimma University Library System (JULS's) users through SDI system by applying an information filtering approach.

Basically, the development of SDI encompasses two main components, documents and users. Thus, during the development of a prototype SDI system, bibliographic lists of books from JULS and information about users from Jimma University website are used. Research design used in this work is Design Science research methodology (DSRM). Python programming language is used for the development of a prototype SDI system. In this study, designing a prototype SDI system involves different methods and techniques such as text operation, indexing, term weighting and document matching which are part of Natural language processing (NLP) and Information retrieval (IR).

After implementing different matching algorithms, the performance of the system is evaluated using recall, precision and F-measure. According to the evaluation result, TFIDF of vector space model performs the best with 70.6% recall, 78.76% precision and 73.76% F-measure. Also the system is validated by library managers scoring about 98% performance. User acceptance test founds 95% of level of agreement to developed prototype SDI system.

Obtained results indicate that the developed prototype SDI system is appreciable to apply in the library environment. However, the system is limited in recommending documents in other languages than English and it needs further investigation. Another drawback of the system is that it registered low performance to long user's information need compared to short user's information need.

Designing SDI system requires careful consideration of user's information need and document representation since the performance of the system is affected by these entities.

In general, just like articles and other documents, it is possible to recommend bibliographic lists of books by applying information filtering mechanism to SDI system.

6.2. Recommendation

Based on system evaluation there are issues that needs further improvement. Therefore, the following points are areas that are left as a future research direction.

- Since this study is done for academic purpose, the developed prototype SDI system is not fully implemented and needs user interface in order to able users interact with the system. Therefore, JULS has to fully implement the SDI system by designing web based user interface in the future since the operation of SDI service is on the internet.
- Performance of SDI system is affected by user information need. So, users should be oriented how to properly set their information need to recommend relevant information that can improve the performance of the SDI system.
- The development of SDI system is language specific since matching between user's information need and document is expressed by the language both information need and document is presented. JULS has collections of different books by different languages. Therefore, SDI system that can recommend different books existed in different languages should be developed in the future.
- In this work, the researcher implemented aged based priority of files to be used as input for SDI system and new file is automatically selected as a source of data from which documents are recommended and stay new until other file is added to directory of files. But if other file is added to files directory before user is recommended the first documents, user losses the information of what is acquired before. To solve this problem we incorporated notification service that tells the delivery of documents on their mail to redirect them to SDI service page. However, due to some circumstances, user may not check his/her email. Therefore, mechanism which captures the last time the user used SDI service and take all file added to file directory after that time as a source of data should be incorporated to the design of this work
- Once the SDI system will be implemented, there is a need to send relevant documents on the mail of the users to notify them immediately the arrival of new books and journals. Even nowadays mobile-based systems are getting more acceptances due to their

efficiency and effectiveness. Such issues need further investigation and left for future researchers.

- In this study, the researcher tried to implement automatic LSI to overcome the problem of polysemy and synonyms. However, due to limited size of corpus/collection, the result is below satisfactory or the result is almost zero recall and precision. Therefore, in the future it is better if thesaurus based semantic modeling approach is incorporate to design SDI system for academic library to solve such problem with similar dataset.

REFERENCE

Ababor, Z. Z. (2003). *Application of Collaborative Filtering Agent for Document Recommendation in SDI System*. Addis Ababa University.

Aberer, K., & De, L. (2006). Information Retrieval and Data Mining Part 1 – Information Retrieval Today ' s Question (pp. 1–31). EPFL-IC.

Altinel, M., & Franklin, M. (2000). Efficient filtering of XML documents for selective dissemination of information. In *Proceedings of the 26th VLDB Conference* (pp. 53–64). Cairo,: VLDB.

Belkin, N., & Croft, W. (1992). Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM*, *29*(10), 1–10.

Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc.

Burke, R. (1999). Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems* (pp. 1–23). Marcel Dekker.

Ceri, S., Bozzon, A., Brambilla, M., Della Valle, E., Fraternali, P., & Quarteroni, S. (2013). Web Information Retrieval (pp. 13–27). Data-Centric Systems and Applications.

Clark, S. (2014). Vector space models of lexical meaning. *Handbook of Contemporary Semantics*, 1–43.

Deerwester, S., Dumais, S. T., Furnas, G. W., & Landauer, T. K. (1990). Indexing by Latent Semantic Analysis. *Journal of the American Society of Information Science*, *41*(6), 391–407.

Felfernig, A., Isak, K., Szabo, K., & Zachar, P. (2007). The VITA financial services sales support environment. *Proceedings of the National Conference on Artificial Intelligence*, *2*, 1692–1699.

González, R. B. (2008). *Index Compression for Information Retrieval Systems*. University of a Coruña

Hanani, U., Shapira, B., & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, *11*(3), 203–259.

Hossain, M. J., & Islam, M. S. (2008). Selective Dissemination of Information (SDI) Service: A Conceptual Paradigm. *International Journal of Information Science and Technology*, *6*(1),

27–44.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems : Principles , methods and evaluation. *Egyptian Informatics Journal*, *2015*(16), 261–273.

JULS. (2014). *Jimma University Library System Annual Report*.

Koubarakis, M., & Koutris, T. (n.d.). Efficient Agent-Based Dissemination of Textual Information. IST Programme of the European Commission

Linden, G., Smith, B., & York, J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing, *7*(1), 1-9

Ling, Y., Wang, X., & Gu, H. (2008). A Hybrid Information Filtering Algorithm Based on Distributed Web Log Mining. *Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on*, *1*, 1086–1091.

Luhn, H. P. (1958). A Business Intelligence System. *IBM Journal*, *2*(4), 314–319.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval*. Cambridge University Press.

Marczyk, G., DeMatteo, D., & Festinger, D. (2005). *Essentials of research design and methodology*. John Wiley & Sons,Inc.

Meteren, R. Van, & Someren, M. Van. (2000). Using Content-Based Filtering for Recommendation. *ECML/MLNET Workshop on Machine Learning and the New Information Age*, 47–56.

Morales-del-Castillo, J. M., Pedraza-Jiménez, R., Peis, E., & Herrera-Viedma, E. (2013). A Semantic Model of Selective Dissemination of Information for Digital Libraries. *Information Technology and Libraries*, *28*(1), 21–31.

Nkiko, C., & Iroaganachi, M. A. (2015). Community-Focused Selective Dissemination of Information Services for Empowering Women Through Information Provision and Utilization: Center for Learning Resouces as a Catalyst for Social Change, 1–14.

O'Neil, E. (2001). Selective Dissemination of Information in the Dynamic Web Environment. Uniersity of Virginia

Oracle and/or its affiliates. (2016). *MySQL Connector/Python Developer Guide*. Oracle and/or its affiliates.

Peffers, K., Tuunanen, T., Rothenberger, M., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information*

Systems, *24*(3), 45–77.

Polyvyanyy, A., & Kuropka, D. (2007). *A Quantitative Evaluation of the Enhanced Topic-Based Vector Space Model*. Hasso-Plattner-Institut für Softwaresystemtechnik an der Universität Potsdam.

Porcel, C., & Herrera-Viedma, E. (2010). Dealing with incomplete information in a fuzzy linguistic recommender system to disseminate information in university digital libraries. *Knowledge-Based Systems*, *23*(1), 32–39.

Porcel, C., Moreno, J. M., & Herrera-Viedma, E. (2009). A multi-disciplinar recommender system to advice research resources in University Digital Libraries. *Expert Systems with Applications*, *36*(10), 12520–12528.

Porcel, C., Tejeda-Lorente, A., Martinez, M. A., & Herrera-Viedma, E. (2012). A hybrid recommender system for the selective dissemination of research resources in a technology transfer office. *Information Sciences*, *184*(1), 1–19.

Rah, J. A., Gul, S., & Wani, Z. A. (2010). University libraries: step towards a web based knowledge management system. *Vine*, *40*(1), 24–38.

Ramos, J., Eden, J., & Edu, R. (n.d.). Using TF-IDF to Determine Word Relevance in Document Queries.

Rao, M. (1993). Selective Dissemination of Information (SDI):A case Study of Central Water and Power Research Station (CWPRS). *Annals of Library Science and Documentation*, *40*(4), 164–154.

Rehurek, R. (2011). *Scalability of Semanticanalysis in Natural Language Processing*. Masaryk University.

Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks* (pp. 45–50). Valletta, Malta: ELRA.

Renganathan, V., Babu, A. N., & Sarbadhikari, S. N. (2013). A Tutorial on Information Filtering Concepts and Methods for Bio-medical Searching. *Journal of Health & Medical Informatics*, *04*(03).

Robin. (2009). Tokenization: Overview. Retrieved September 22, 2016, from http://language.worldofcomputing.net/category/tokenization

Rosario, B. (2000). Latent Semantic Indexing : An overview. *Infosys 240*, 1–16.

Shuda, W., Jiangping, L., & Riu, W. (2009). Research of Information Filtering Based on Vector Space Model. *2009 Second International Workshop on Computer Science and Engineering*, 42–46.

Singh, V., & Saini, B. (2014). An Effective Pre-Processing Algorithm for Information Retrieval Systems. *International Journal of Database Management Systems (IJDMS)*, *6*(6), 13–24.

Spiegel, S. (2009). *A Hybrid Approach to Recommender Systems based on Matrix Factorization presented by*. *Technical University Berlin*. Technical University Berlin.

Tutorials Point. (2014). *Python Programming Language*. Tutorials Point (I) Pvt. Ltd.

tutorialspoint. (2016). Modules and Packages. Retrieved October 8, 2016, from https://www.tutorialspoint.com/python/python_modules.htm

Vembunarayanan, J. (2013). Tf-Idf and Cosine similarity. Retrieved July 15, 2016, from https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/

Wang, X., Xie, Y., & Li, B. (2006). A hybrid information filtering model. In *2006 International Conference on Computational Intelligence and Security, ICCIAS 2006* (pp. 1049–1054). IEEE.

Wood, M. S., & Seeds, R. S. (1974). Development of SDI services from a manual current awareness service to SDILINE. *Bulletin of the Medical Library Association*, *62*(4), 374–384.

Yan, T. W., & Garcia-molina, H. (n.d.). Index Structures for Information Filtering Under the Vector Space Model, 1–11.

Zhang, H. R., Min, F., He, X., & Xu, Y. Y. (2015). A hybrid recommender system based on user-recommender interaction. *Mathematical Problems in Engineering*, *2015*(1), 1-12

APPENDICES

Appendix I: Prototype evaluation questionnaire for library managers

Dear JULS manager, this questionnaire is provided to be filled by you to evaluate the performance of prototype SDI system that you looked. I would like to thank you in advance for your interest and valuable time.

Instruction: Please, tick your option to the side of each evaluation parameter under performance value.

No	Evaluation Parameters	Performance Value				
		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	Proposed system can solve the problem at hand in the library					
2	The system is easy to use					
3	The system is efficient in time and memory					
4	The system is significant in the library					

Appendix II: Prototype evaluation questionnaire for library users

Dear JULS user, this questionnaire is provided to be filled by you to evaluate the performance of prototype SDI system that you looked. I would like to thank you in advance for your interest and time.

Instruction: Please, tick your option to the side of each evaluation parameter under performance value.

No	Evaluation Parameters	Performance Value				
		Strongly disagree	Disagree	Neutral	Agree	Strongly agree
1	The prototype system presented all relevant items for me					
2	I found all relevant items to my subject interest on the top					
3	I recommended documents as soon as I insert my User_Id					
4	No more effort is required from me while I use the system to obtain relevant document to my subject interest					
5	Recommended items are presented in easy way that I can read and understand					
6	Application of such system is very necessary in the library					

Appendix III: Precision and recall along with relevance judgment of top ten recommended documents for string-matching model

User Id	Document in their ranking	Relevance judgment	Recall	Precision
JU/PHMS/10	Food Microbiology	R	0.1	1
	Food Microbiology: A laboratory manual	R	0.2	1
	Laboratory Methods in Food microbiology	R	0.3	1
	Microbiology Principles and Explorations	R	0.4	1
	Modern Food Microbiology	R	0.5	1
	Manual of Clinical Microbiology	R	0.6	1
	Medical Microbiology	R	0.7	1
	Medical Microbiology (by other author)	R	0.8	1
	Sherris Medical Microbiology	R	0.9	1
	Food Microbiology (duplicated)	R	1	0.9
		Average	1	1
JU/IT/03	A Complete Hospital Manual of Instruments and Procedures	R	0.1	1
	Handbook of Food Analysis Instruments	R	0.2	1
	Handbook of Food Analysis Instruments (duplicated)	NR	0.2	0.67
	Ultrasound physics and instrumentation	R	0.3	0.75
	CAPITAL MARKET:INSTITUTIONS AND INSTRUMENTS	NR	0.3	0.6
	A course in Electrical and Electronics Measurements and instrumentation	R	0.4	0.67
	A Course in Electronics and electrical measurements and instrumentation	NR	0.4	0.57
	Analysis and Application of Analog Electronic Circuits to Biomedical Instrumentation	R	0.6	0.625

	Ultrasound physics and instrumentation (2 vol.set)	NR	0.6	0.6
		Avrage	0.56	0.72
JU/CNS/15	Collection development in the digital age	R	0.1	1
	Describing Electronic, Digital and Other media Using AACR2 and RDA: How to-do-it Manual and CD-ROM for Librarians (with CD-ROM)	R	0.2	1
	Digital Archives: Management, access and use	NR	0.2	0.67
	Digital Inclusion: Measuring the Impact of Information and Community Technology	R	0.3	0.75
	Digital information design and Access	R	0.4	0.8
	Digital Libraries and information Access: Research Perspectives	R	0.5	0.83
	Digital preservation	R	0.6	0.857
	Digitizing Collections: Strategic Issues for the information manager	R	0.7	0.875
	Exploring Digital Libraries: Foundations, Practice, Prospects	R	0.8	0.89
	Information Representation and Retrieval in the Digital Age	R	0.9	0.9
		Average	0.9	0.86
JU/CAVM/03	Alternative Building Materials & Technologies	NR	0	0
	CURRICULUM: ALTERNATIVE APPROACHES, ONGOING ISSUES	NR	0	0
	Experiments in Unit Operations and Processing of Foods	R	0.1	0.33
	Food Processing Technology: Principles and Practice	R	0.2	0.5
	Fruit and Vegetable Processing: Improving Quality	R	0.3	0.6
	Handbook of vegetable preservation and processing	R	0.4	0.67
	Unit Operations and Processes in Environmental Engineering	NR	0.4	0.57

	Dynamic Information and Library Processing	NR	0.4	0.5
	Extensible Processing for Archives and special collections	NR	0.4	0.44
	PRACTICAL BATCH PROCESS MANAGEMENT	NR	0.4	0.4
		Average	0.4	0.41
JU/IT/12	Ecological Models and Data in R.	NR	0	0
	Computing and Mathematical Modeling	NR	0	0
	MODELLING NON-STATIONARY ECONMIC TIME SERIES: A MULTIVARIATE APPROACH	NR	0	0
	MODELING AGGREGATE BEHAVIOR AND FLUCTUATIONS INCCONMICS: STOCHASTIC VIEWS OF INTERA.	NR	0	0
	INCOME DISTRIBUTION IN MACROECONMIC MODELS	NR	0	0
	ECONMETRIC MODELING: A LIKELIHOOD APPROACH	NR	0	0
	Biomedical Signal Processing & Signal Modeling	NR	0	0
	Experiments in Unit Operations and Processing of Foods	NR	0	0
	Food Processing Technology: Principles and Practice	NR	0	0
	Fruit and Vegetable Processing: Improving Quality	NR	0	0
		Average	0	0
JU/CSSH/02	GENERATIVE-SOCIOLINGUISTIC PERSPECTIVE ON CONCORD VARIATION	R	0.1	1
	adaptive physical education and Sport	NR	0.1	0.5
	administration of physical education and Sport Program	NR	0.1	0.33
	biomechanics in physical education	NR	0.1	0.25
	CURRICULUM 21: ESSENTIAL EDUCATION FOR A CHANGING WORLD	NR	0.1	0.2

	curriculum design in physical education	NR	0.1	0.17
	DECENTRALIZATION OF EDUCATION: TEACHER MANAGEMENT	NR	0.1	0.14
	Education and sports psychology	NR	0.1	0.125
	EDUCATIONAL PSYCHOLOGY (WITH MYEDUCATIONLAB)	NR	0.1	0.1
	EDUCATIONAL RESEARCH: COMPETENCIES FOR ANALYSIS AND APPLICATIONS	NR	0.1	0.1
		Average	0.1	0.29
JU/CLG/01	Cataloging and decision making in hybrid Environment: The translation from AACR2 to RDA	NR	0	0
	ENVIRONMENT AND STATECRAFT: THE STRATEGY OF ENVIRONMENTAL TREATY-MAKING,	R	0.1	0.5
	Food Product Development: From Concept to the Marketplace	NR	0.1	0.33
	Food product development: Maximizing success	NR	0.1	0.25
	Functional Food Product Development	NR	0.1	0.2
	Ten Steps to a Results-Based Monitoring and Evaluation System: A Handbook for Development Practitioners	R	0.2	0.33
	Advances in Mathematics Scientific Developments and Engineering Applications	NR	0.2	0.28
	Collection development in the digital age	NR	0.2	0.25
	Customer-based collection development: An Overview	NR	0.2	0.22
	Fundamental of collection development and Management	NR	0.2	0.2
		Average	0.2	0.256
JU/PH	A Comparison of Urinalysis Technologies for Drug	R	0.1	1

MS/28	Testing in Criminal Justice			
	Herbal Drugs and hytopharmaceuticals	R	0.2	1
	Neonatology: Management, Procedures, On-Call Problems, Diseases, and Drugs	R	0.3	1
	Clinician's Pocket Drug Reference 2009	R	0.4	1
	Neonatology: Management, Procedures, On-Call Problems, Diseases, and Drugs (duplicate)	NR	0.4	0.8
	A Comparison of Urinalysis Technologies for Drug Testing in Criminal Justice (duplicate)	NR	0.4	0.67
	Manual of Laboratory & Diagnostic Tests	R	0.5	0.71
	Information Retrieval Systems: Characteristics, Testing, and Evaluation (Information Science)	NR	0.5	0.625
	PSYCHOLOGICAL TESTING AND ASSESSMENT	NR	0.5	0.56
	A Comparison of Urinalysis Technologies for Drug Testing in Criminal Justice (duplicate)	NR	0.5	0.5
		Average	0.5	0.79
JU/BECO/03	ENTREPRENEURSHIP AND LOCAL ECONMIC DEVELOPMENT	R	0.1	1
	ENTREPRENEURSHIP, COMPETITIVENESS AND LOCAL DEVELOPMENT: FRONTIERS IN EUROPEAN ENTREPRENEUR.	R	0.2	1
	YOUTH ENTREPRENEURSHIP AND LOCAL DEVELOPMENT IN CENTRAL AND EASTERN EUROPE	R	0.3	1
	RESEARCH IN ORGANIZATIONAL BEHAVIOR 1997: AN ANNUAL SERIES OF ANALYTICAL.	R	0.4	1
	ORGANIZATIONAL SURVIVAL IN THE NEW WORLD	NR	0.4	0.8
	ORGANIZATIONAL BEHAVIOR	R	0.5	0.83
	ORGANIZATIONAL CHANGE AND	R	0.6	0.86

	DEVELOPMENT CONTROL SYSTEMS PROCESS INNVATION			
	ORGANIZATIONAL BEHAVIOUR	R	0.7	0.875
	ORGANIZATIONAL BEHAVIOUR	NR	0.7	0.78
	ORGANIZATIONAL BEHAVIOUR	NR	0.7	0.7
		Average	0.7	0.8845
JU/CNS/04	A Comparison of Urinalysis Technologies for Drug Testing in Criminal Justice	NR	0	0
	Herbal Drugs and hytopharmaceuticals	R	0.1	0.5
	Neonatology: Management, Procedures, On-Call Problems, Diseases, and Drugs	R	0.2	0.67
	Clinician's Pocket Drug Reference 2009	R	0.3	0.75
	Neonatology: Management, Procedures, On-Call Problems, Diseases (duplicated)	NR	0.3	0.6
	Handbook on Impact Evaluation: Quantitative Methods and Practices (World Bank Training Series)	NR	0.3	0.5
	Digital Inclusion: Measuring the Impact of Information and Community Technology	NR	0.3	0.43
	Evaluating the Impact of Your Library	NR	0.3	0.375
	HARRISONS Infectious Diseases	R	0.4	0.44
	Principles and Practice of Pediatric Infectious Disease	R	0.5	0.5
		Average	0.5	0.51

Appendix IV: Precision and recall along with relevance judgment of top ten recommended documents for TFIDF model

User Id	Documents in their ranking	Relevance judgment	Recall	Precision
JU/PHMS/10	Food Microbiology	R	0.1	1
	Food Microbiology	R	0.2	1
	Medical Microbiology	R	0.3	1
	Medical Microbiology	R	0.4	1
	Sherris Medical Microbiology	R	0.5	1
	Modern Food Microbiology	R	0.6	1
	Microbiology Principles and Explorations	R	0.7	1
	Laboratory Methods in Food microbiology	R	0.8	1
	Food Microbiology: A laboratory manual	R	0.9	1
	Manual of Clinical Microbiology	R	1	1
		Average	1	1
JU/IT/03	Handbook of Food Analysis Instruments	R	0.1	1
	Handbook of Food Analysis Instruments	R	0.2	1

	Analysis and Application of Analog Electronic Circuits to Biomedical Instrumentation	R	0.3	1
	CAPITAL MARKET:INSTITUTIONS AND INSTRUMENTS	NR	0.3	0.75
	A course in Electrical and Electronics Measurements and instrumentation	R	0.4	0.8
	A Course in Electronics and electrical measurements and instrumentation	R	0.56	0.83
	Ultrasound physics and instrumentation	R	0.67	0.857
	Ultrasound physics and instrumentation	R	0.78	0.875
	A Complete Hospital Manual of Instruments and Procedures	R	0.89	0.89
		Average	0.89	0.8891
JU/CNS/15	Digital Libraries and information Access: Research Prespectives	R	0.1	1
	Exploring Digital Libraries: Foundations, Practice, Prospects	R	0.2	1
	Digital preservation	R	0.3	1
	Biomedical Digital signal processing	NR	0.3	0.75
	Digital information design and Access	R	0.4	0.8
	Collection development in the digital age	R	0.5	0.83
	Information Representation and Retrieval in the Digital Age	R	0.6	0.857
	Digitizing Collections: Strategic Issues for the information manager	R	0.7	0.875
	Digital Archives: Management, access and use	R	0.8	0.89
	Essential Library of congress subject headings	R	0.9	0.9
		Average	0.9	0.8902
JU/CAVM/03	Fruit and Vegetable Processing: Improving Quality	R	0.1	1
	Food Processing Technology: Principles and Practice	R	0.2	1

	Chemical Process Safety Fundamentals with Applications	NR	0.2	0.67
	Chemical Process Safety: Fundamentals with Applications	NR	0.2	0.5
	SIX SIGMA : CONTINUAL IMPROVEMENT FOR BUSINCSS	NR	0.2	0.4
	Quality in Laboratory Hemostasis and Thrombosis	R	0.3	0.5
	PRACTICAL BATCH PROCESS MANAGEMENT	NR	0.3	0.43
	Engineering Properties of Foods	R	0.4	0.5
	Food Properties Handbook	R	0.5	0.56
	Physical Properties of Foods	R	0.6	0.6
		Average	0.6	0.616
JU/IT/12	Irrigation Water Power And Water Resources Engineering	R	0.1	1
	Irrigation Water Management Principles And Practice	R	0.2	1
	Irrigation Water Power Engineering	R	0.3	1
	ECONOMETRIC MODELING: A LIKELIHOOD APPROACH	NR	0.3	0.75
	Computing and Mathematical Modeling	NR	0.3	0.6
	Ecological Models and Data in R	NR	0.3	0.5
	INCOME DISTRIBUTION IN MACROECONMIC MODELS	NR	0.3	0.43
	HUMAN RESOURCE MANAGEMENT	NR	0.3	0.375
	HUMAN RESOURCE MANAGEMEN	NR	0.3	0.33
	HUMAN RESOURCE MANAGEMENT	NR	0.3	0.3
		Average	0.3	0.6285
JU/CSSH/02	LANGUAGE OF SPEECH AND WRITING	R	0.1	1
	LANGUAGE OF THE NEWS	R	0.2	1
	LANGUAGE OF WAR	NR	0.2	0.67
	LANGUAGE OF COMICS	R	0.3	0.75
	LANGUAGE OF CONVERSATION	R	0.4	0.8
	LANGUAGE OF DRAMA	R	0.5	0.83
	LANGUAGE OF HUMOUR	NR	0.5	0.71
	LANGUAGE OF MAGAZINES	R	0.6	0.75
	LANGUAGE OF POETRY	R	0.7	0.78
	LANGUAGE OF TELEVISION	R	0.8	0.8
		Average	0.8	0.809
JU/CL	DEVELOPMENT OF ECONMIC ANALYSIS	R	0.1	1

G/01	DEVELOPING THE CURRICULUM	NR	0.1	0.5
	RURAL WORLD, THE: EDUCATION AND DEVELOPMENT	R	0.2	0.67
	PRODUCT DESIGN AND DEVELOPMENT	R	0.3	0.75
	Fundamental of collection development and Management	NR	0.3	0.6
	Functional Food Product Development	NR	0.3	0.5
	Customer-based collection development: An Overview	NR	0.3	0.43
	Cataloging and decision making in hybrid Environment: The translation from AACR2 to RDA	NR	0.3	0.375
	Advances in Mathematics Scientific Developments and Engineering Applications	NR	0.3	0.33
	ENVIRONMENT AND STATECRAFT: THE STRATEGY OF ENVIRONMENTAL TREATY-MAKING	R	0.4	0.4
		Average	0.4	0.5555
JU/PHMS/28	A Comparison of Urinalysis Technologies for Drug Testing in Criminal Justice	R	0.11	1
	Herbal Drugs and hytopharmaceuticals	R	0.22	1
	ETHICAL ISSUES FOR ESL FACULTY : SOCIAL JUSTICE IN PRACTICE	NR	0.22	0.67
	PSYCHOLOGICAL TESTING AND ASSESSMENT	NR	0.22	0.5
	Manual of Laboratory & Diagnostic Tests	R	0.33	0.6
	Clinician's Pocket Drug Reference 2009	R	0.44	0.67
	Information Retrieval Systems: Characteristics, Testing, and Evaluation (Information Science)	NR	0.44	0.57
	Neonatology: Management, Procedures, On-Call Problems, Diseases, and Drugs	R	0.56	0.625
	Neonatology: Management, Procedures, On-Call Problems, Diseases, and Drugs	R	0.67	0.67
		Average	0.67	0.7005555556
JU/BECO/03	Leadership and Nursing Care Management	R	0.1	1
	SUPERVISION AND INSTRUCTIONAL LEADERSHIP: A DEVELOPMENTAL APPROACH	R	0.2	1

	HUMAN RESOURCE MANAGEMENT	R	0.3	1
	HUMAN RESOURCE MANAGEMENT	R	0.4	1
	HUMAN RESOURCE MANAGEMENT	R	0.5	1
	HUMAN RESOURCE MANAGEMENT	R	0.6	1
	Cultural Heritage information Access and Management	NR	0.6	0.857
	HUMAN RESOURCE MANAGEMENT	R	0.7	0.875
	INTRODUCTION TO HUMAN RESOURCE MANAGEMENT	R	0.8	0.89
	OPERATIONS MANAGEMENT	R	0.9	0.9
		Average	0.9	0.9522
JU/CNS/04	HARRISONS Infectious Diseases	R	0.1	1
	Herbal Drugs and hytopharmaceuticals	R	0.2	1
	Principles and Practice of Pediatric Infectious Disease	R	0.3	1
	Principles and Practice of Pediatric Infectious Disease	R	0.4	1
	Evaluating the Impact of Your Library	NR	0.4	0.8
	Clinician's Pocket Drug Reference 2009	R	0.5	0.83
	Digital Inclusion: Measuring the Impact of Information and Community Technology	NR	0.5	0.71
	Immunology & Serology in Laboratory Medicine	R	0.6	0.75
	comprehensive study of sports medicine	NR	0.6	0.67
	Handbook on Impact Evaluation: Quantitative Methods and Practices (World Bank Training Series)	NR	0.6	0.6
		Average	0.6	0.836

Appendix V: Lists of total population and sample taken by departments

Department name	Population	Samples taken	Samples in %
Accounting	23	2	8.7
Banking	7	2	28.57
Economics	22	2	9.1
Management	25	2	8
Horticulture and Plant sciences	22	2	9.1
Postharvest Management	10	2	20
Animal Science	18	2	11.1
Natural Resources Management	34	2	5.9
Agriculture Economics, Agribusiness and Rural Development	25	2	8
Veterinary Medicine	23	2	8.7
Governance and Development Studies	27	2	7.4
Law	26	2	7.7
Biology	24	2	8.3
Chemistry	25	2	8
Information Science	22	2	9.1
Mathematics	23	2	8.7
Physics	15	2	13.3
Sport Science	6	2	33.3
Statistics	20	2	10
Afan Oromo	12	2	16.67
Amharic Language and Literature	12	2	16.67
English Language and Literature	44	2	4.5
Geography and Environmental Studies	10	2	20
History	11	2	18.2
Oromo Folklore	22	2	9.1
Sociology	16	2	12.5

Biomedical Sciences (health)	47	2	4.3
Epidemiology	20	2	10
Health Education	22	2	9.1
Health service management	8	2	25
Internal medicine	14	2	14.3
Medical Laboratory Sciences and Pathology	31	2	6.5
Pharmacy	26	2	7.7
Population and Family Health	7	2	28.6
Anesthesia	20	2	10
Dentistry	10	2	20
Environmental Health Sciences and Technology	29	2	7
Civil and Environmental engineering	82	2	2.4
Biomedical Engineering	18	2	11.1
Chemical Engineering	16	2	12.5
Electrical and Computer Engineering	18	2	11.1
Mechanical Engineering	16	2	12.5
Water resources and Environmental Engineering	13	2	15.4
Total	921	86	9.33