

A Framework for Mobile Based Research Paper Recommendation in a Conference

Aklilu Mandefro Messele

Department of Computer Science, Volunteer Tech®, Gondar, Ethiopia

Email address:

aklilu.mandefro@volunteer-tech.com

To cite this article:

Aklilu Mandefro Messele. A Framework for Mobile Based Research Paper Recommendation in a Conference. *International Journal of Data Science and Analysis*. Vol. 8, No. 5, 2022, pp. 131-148. doi: 10.11648/j.ijdsa.20220805.11

Received: June 14, 2022; **Accepted:** July 19, 2022; **Published:** September 29, 2022

Abstract: Finding conferences with papers relevant to their interests can be difficult for research conference attendees because everyone has different preferences. To address this issue, this research proposes a framework and a prototype of a personalized recommendation system for research conference items. When making recommendations, the prototype considers the user's research area and college. The prototype employs three algorithms to recommend conference papers based on what users have previously read: a collaborative filtering algorithm (k-Nearest Neighbor), a content-based filtering algorithm, and a hybrid of the two. The design science research paradigm was used to write the research. This research covers the conceptual framework design and prototype implementation in programming languages that the researcher is capable of implementing, as well as a brief state of the art of the recommending systems literature. The prototype's usability was assessed using the information retrieval concept. To assess the quality of recommendations, system performance and a user-centered evaluation were performed. The usability evaluation results showed that users were generally pleased with the prototype's usability. Users who tested the prototype were generally pleased with the quality of the recommendations. The performance of a prototype system is 86 percent, and user acceptance is 86.5 percent. Finally, future works in the area are clearly stated.

Keywords: Research Conference, Mobile Based Systems, SMO Classifier, Framework, Android, Recommender System

1. Introduction

1.1. Background

A vigorous inquiry and systematic investigation that contributes to a field's knowledge base is referred to as research. It serves as a sound foundation for making decisions about policies, plans, and actions [46]. The goal of research is to "go beyond what is already known in the physical, biological, or social world." Research differs from other forms of knowledge discovery (such as reading a book) in that it employs a systematic process known as the Scientific Method [7].

Research has a positive impact on socio economic development [8, 9]. According to the UK Department for International Development (UFID), four paths are commonly cited to describe how the research will contribute to development: Investment in research will drive economic growth, increase human capital, lead to the development of pro-poor products and technologies, and provide evidence to

inform policies and practice.

When research organizations want to collaborate with multiple partners to achieve specific goals in national or international development, they issue a call for research proposals via a notice on their website or handwritten letters. A research call invites interested parties to submit proposals for international development initiatives that will help achieve the desired results [47].

The call for proposals seeks high-quality research that is in line with the countries and organizations' research priorities and inquiries. These research areas were determined through extensive consultation with policymakers and are intended to fill existing research gaps and policy questions [32]. It invites university leaders, academics, researchers, students, policymakers, practitioners, and all other interested educationalists and business people from around the world to participate and submit research articles or article proposals along the thematic lines specified in announcements [65].

Each institution's research organizing committee invites the submission of abstracts for papers, papers, or posters. The abstracts for proposed research should make clear (as appropriate to the type of inquiry) the educational context to which the paper relates, the type of inquiry or methodology, the central research question(s) or hypo research work, and the paper's contribution to education for development, as well as key bibliographic references to evaluate and propose for the final selection.

It is not an easy task to recommend personalized conference papers to users [40]. As a result, we proposed a hybrid research conference papers recommendation framework that takes into account additional user and paper characteristics. We propose phones as dissemination devices to end users because we live in a mobility era in which mobile technologies allow users to move around. Phones are being used for more than just voice transmission; mobile applications such as location-based services, mobile-commerce applications, and item recommendations are being deployed at an increasing rate [3, 44].

People can share their opinions and learn from each other's experiences using recommender systems [4]. This is based on user information, Meta data associated with items, and/or implicit or explicit user ratings for items [2]. Recommender systems all have a way to describe the items that might be recommended, a way to create a user profile that describes the types of items the user likes, and a way to compare items to the user profile to determine what to recommend [5, 11]. According to the same author, mobile multimedia recommender systems typically perform three primary functions: information collection (explicit or implicit feedback), recommendation learning (learning algorithm and information filtering), and resource prediction/recommendation. Mobile multimedia recommender systems collect all user interests pertaining to multimedia information for the prediction task using explicit and implicit feedback methods, including the users' attributes, behaviors, the content of the resources the mobile user accesses, as well as context such as time, location, and other social and behavioral contexts [60]. MTRS, a web-to-mobile tourist framework, discovers that implicit user modeling is more accurate, dependable, and non-intrusive than explicit user modeling, and it employs a clustering algorithm to classify users with similar interests [18].

1.2. Statement of the Problem

One of the main goals of the research conference strategy is to teach and transform research trends to participants so that the necessary information can be found in less time and with less effort. Many governments around the world have spent significant time and effort promoting research as a policy [14]. Governments typically provide information about research entities such as current issues, research directions, research trainings, and activities in action and practice. This information can help citizens and international researchers get better services [28].

According to the reviewed literature, existing research

conferences are primarily concerned with providing call for proposal information online to authors about the content and evaluation of their papers [30, 49]. It does not provide participants with any information. It is difficult to provide personalized recommendation service because research calls for papers are not based on systems. As a result, it is up to users or participants to find research conferences that interest them. And disseminating information about the conference and the papers that will be on display is extremely difficult [39].

According to research conducted at various public universities, research calls are typically announced via university official websites and social media platforms such as Facebook [45, 58]. As a result, users and authors must visit all higher education and research institute websites. Users can learn about the conference schedule thanks to the proliferation of websites for universities and other research organizations. However, they are not informed about the conference articles that will be displayed at the exhibition. Many people find it exhausting to visit websites in search of something that piques their interest [19]. This means that users visit far more websites than they can handle, potentially missing out on programmes that would be of interest to them.

Many previous mobile recommender systems have concentrated on points of interest, travel and tourism, and media [13, 31]. However, no work in the area of mobile-based recommendation on conference presentation papers has been published. A multidimensional GUI increases user acceptance and confidence in recommendations by serving as an interface to guide the recommender's criteria in selecting an item or user neighborhoods [10].

It is widely accepted that in order to make research call services more appealing to users, calling must be delivered in a user-centered manner [12, 35]. In this way, research organizations can improve the delivery of their research calling services on a personalized basis, ensuring that the needs and interests of heterogeneous users are met without requiring excessive data input from users. The framework created is used to provide users with the best research resources and presentations that can meet their information needs at the conference by utilizing contexts such as user preferences and item meta-data and also putting them into contextual context. Papers to be presented at the conference are chosen as usual, but we are working on a framework to recommend users based on the committee's recommendations.

Therefore, this research focused on answering the following questions.

1. How to incorporate users' research interest in recommendation systems?
2. How to design a conceptual framework that can be used to support a potential research conference papers recommendation and to provide personalization for a specific research domain?
3. How to evaluate and improve the performance of the recommender system.

2. Experimental Procedures, Materials, and Methods

2.1. Data Sources and Corpus Preparation

Primary data were collected and prepared from the University of Gondar and Bahirdar University research center, which leads the research call and presentation. The researchers have prepared these unstructured data to structured and semi-structured formats. The researchers produced labeled arff file, binary arff file and tfidf arff data (word document vector represented data) for experimentation. This unstructured textual corpus has been preprocessed before we used it for our proof-of-concept system representation of the designed recommendation framework.

2.2. Recommendation Technique and Tools

In this study, the researchers used different subject category Content classification techniques to filter 'research conference papers' as an item. Document preprocessing and representation were employed to prepare papers as item to users. Most common indicators Term Frequency (TF), Inverse Document Frequency (IDF) and their multiplicative combination $TF \times IDF$ are employed. Then, the most effective document categorization technique is applied. There are many approaches to develop recommendation system such as: content based, collaborative filtering, demographic, knowledge based, community based and hybrid recommendation systems [5, 53]. From those techniques, a hybrid recommender systems approaches have been selected.

For developing the prototype, Android OS as application running environment, Eclipse/ Android as development IDE and PHP as server-side script has used.

2.3. Research Methodology

2.3.1. Research Design

This study was conducted using the Design Science Research methodology. Design science seeks to create innovations that define the ideas, practices, technical capabilities, and products that can be used to effectively and efficiently analyze, design, implement, and use information systems. Hevner et al. [25] present a framework with a set of seven guidelines for conducting design science research. Design science revolves around the creation of an artifact and its rigorous evaluation [61]. An artifact was created and evaluated as part of this research project.

2.3.2. Document Representation and Classification Techniques

(i). Text Preprocessing

Text preprocessing techniques such as tokenization, stemming, and stop word removal are used to preprocess the paper documents. Boolean weighting, term frequency weighting, and term frequency inverse document frequency weighting are the most common term weighting approaches

used in text categorization. The simplest method of term weighting is Boolean weighting, which assigns 1 if the term appears in a document and 0 if the term does not appear in the document [37, 64]. The weighting only considers the presence or absence of the term in the document and does not specify which documents the term appears more or less frequently. Term frequency weighting takes into account the frequency with which the term appears in documents. The weight of a term in a document in this term weighting is equal to the number of times the term appears in the document. Sometimes the most common term cannot distinguish one document from another. If a term's term frequency is high, its discriminating power against mean documents is low. As a result, this term weighting technique is rarely employed in text categorization processes. The frequency of the most discriminating term of documents is used in term frequency inverse document frequency weighting (tfidf).

$$Tf(ti, Dj) = f(ti, Dj) = \max(f(ti, Dj)) \quad (1)$$

$$idf(ti) = \log \left(\frac{\text{Total number of research articles}}{\text{Number of research articles contain term } ti} \right) \quad (2)$$

$$w(ti, Dj) = Tf(ti, Dj) * idf(ti) \quad (3)$$

Where $Tf(ti, Dj)$ is the term frequency of term ti in article dj , $f(ti, Dj)$ is the frequency of term ti in article Dj , $idf(ti)$ is the inverse document frequency of term ti , and $w(ti, Dj)$ is the TF-IDF weight of term ti in article Dj .

(ii). Vector Space Model

The VSM consists of three parts [17]:

Document Indexing is the main reason for the efficient similarity check, between query and documents, it can be thought of as the index in the back of a book, indicating what pages a word can be found on. In the same analogy; for query to document matching, would be to read every word on every page to find the related pages.

Term Weighting is the act of assigning a value to a term in a document, which indicates the measure of relevance to that term in that document [54]. The most common version of term weighting in search engines is called Term Frequency - Inverse Document Frequency.

Similarity is the measure of similarity between a document and a query. The most common version of this measure is called cosine similarity.

(iii). Classification

Classification techniques decide how much a thing is or isn't part of some type or category, or how much it does or doesn't have some attribute [68]. Often these systems learn by reviewing many instances of items in the categories in order to deduce classification rules. Classification helps decide whether a new input or thing matches a previously observed pattern or not, and it's often used to classify behavior or patterns as unusual [15]. The documents are classified using the support vector machine classifier (SMO). A SMO classifier predicts the target value of the class based on various attributes of the data set. The researchers use the

SMO classifier for classifying documents data set.

2.3.3. Hybridization Methods

Each recommender system technique has its own set of challenges, benefits, and drawbacks. Significant research effort has gone into hybrid recommendation methods that combine collaborative and content-based filtering, as well as other recommendation methods [5, 14]. The following are the various hybridization strategies and methods for hybrid recommender systems [66]:

- 1) *Switching*: Depending on the current situation, the system switches between recommendation techniques. It switches between recommendation techniques based on some criterion.
- 2) *Mixed*: recommendations from multiple recommenders are displayed at the same time. Recommendations from multiple techniques are presented together, but they are not hybridized [38].
- 3) *Weighted*: several recommendation techniques' scores (or votes) are combined to produce a single recommendation. The utility of an item is calculated by combining the outcomes of all available recommendation techniques.
- 4) *Cascade*: one recommender refines another's recommendations. Recommendation techniques are used sequentially, breaking ties and refining results from the previous technique. The output of one recommender does not become the input for the next, but the results of the recommenders involved are prioritized and combined.
- 5) *Feature combination*: features from various recommendation data sources are combined into a single recommendation algorithm.
- 6) *Meta-level*: one recommender's model is used as input

to another.

- 7) *Feature augmentation*: one technique's output is used as an input feature for another.

The weighted hybrid recommender hybridizing technique is used by the researchers. A recommended item's score is computed using the results of all available recommendation techniques in the system (i.e. CBF/content-based filtering and CF/collaborative filtering/).

2.4. Dataset

One of the important points before generating the recommendations is the dataset. Depending on the dataset it could have very different results, if the data is too short; it is possible to have non accurate recommendations. A dataset created for the purpose of this research is using data from university of Gondar and Bahirdar university research services (see table 1 and figure 1). The data was stored in flat electronically and collected manually as they have not publicly available API. The database stores the article text and their meta-data, user profiles, ratings and predictions.

Table 1. Dataset properties.

Property	Column
Users	User
Authors	author
Research Conferences	provider
Conference participation	conference_participation
User similarities	Usersimilarity
Location	Location
Papers	Paper
Category	Category
Rate	Rate

The dataset was created using a MYSQL database with the schema seen in Figure 1 below.

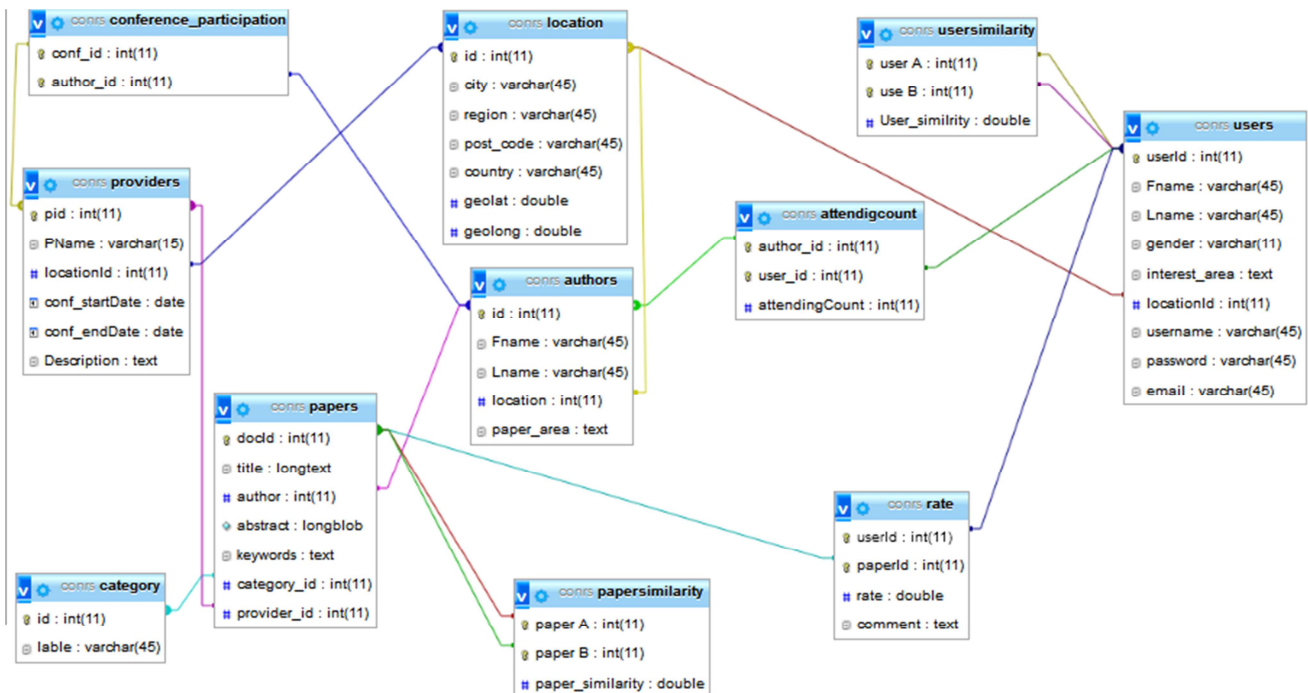


Figure 1. Conference Recommendation Dataset.

2.5. Implementation Tools and Technology

To develop the prototypical representation for the designed framework, the researchers used the list of tools given in table 2.

Table 2. Implementation tools.

Category	Tools
Target platform	Android OS
Source code	Java
Development IDE	Eclipse ADT
Server-side script	PHP
PHP Editor	Notepad++
Connect mobile app with server	JSON

2.5.1. Framework Development

Research call services must be proactive in providing information and tailoring their services to users.

Creating a framework for automatic recommender system (RS) generation [44] is one way to make RS development more independent of software developers, and thus reduce cost and development time. The first step in creating this framework is to create a domain-independent model that represents information about users (participants) and items (papers for research conference presentations). These models must be general enough to capture data from real-world applications. As a result, the goal of this framework is to identify a set of information related to research conference participants and to recommend items (papers) in relation to conferences held at specific times.

Personalized service is defined as the process of gathering information from web users and using that information to tailor services to their needs and preferences [44]. This framework is divided into three basic phases: data collection (a collection of papers that have passed peer review evaluation), data processing (classifying papers into their specific-domain subject category), and delivery of

personalized N-top papers. Finally, if more than one organization is calling for papers at the same time, the user weights the paper relevancy based on his/her own criteria and selects a research conference.

A research call for papers can improve the process and management of academic and research, which correlates strongly with achieving better problem solutions by increasing the convenience and accessibility of current issues. On the other hand, large amounts of information from research communities can cause serious information overload issues and impede the effectiveness of providing research call services. In such cases, participants looking for research conferences to attend struggle to find useful information about the conference and become increasingly frustrated by the difficulties in locating the right information about it and its corresponding services. Other than the work of some scientific paper recommender systems, there is no widely accepted framework for personalization in the context of research conference presentation paper recommendation.

The proposed framework aims to help research participants by providing relevant papers present in the research conference based on their profile information.

2.5.2. Design Requirements

The following scenario describes how a user preference-based research conference presentation paper recommendation could be used. A user wants to attend some duration of days for research paper presentation in the research conference. Participant's preference in research idea is quite different as of motivations or incentives those conferences provide, so choosing what conference to attend is a challenging task. They would love an application that would give recommendations for which research conference for papers to attend based on what sort of specified area they have read / write / specialize before, within a specified time period.

Table 3. Design requirements.

Requirement
1. Research Conference papers recommendations should be based on a user's research preference
2. The system should consider conference providers and participants so that users can aware of it.
3. The system should take time into account; it should not recommend conferences that already have taken place or conferences too far ahead in time.

From this scenario, three requirements for the final prototype are extracted as seen in the above table. The prototype implements the proposed framework following its design, to improve the usability of the framework.

(i). Design Approach

In a recommendation process for research conference papers, P ; a set of conferences, C ; the authors/presenters, A ; and a set of users, U ; are used to provide research conference paper recommendation for a user u . Each of the conferences has a set of authors presenting. Each of the users in U has given a rating to a subset of the papers [29]. A set of context parameters like preference of a user, and time are defined where the conference items recommended should adhere to.

The goal of the recommendation system is to provide the top N most relevant papers $P' \subseteq C$ for u in each provider as a whole in the context applied, CP , and the ratings, R , each of u previously have given [21, 27].

In this research, three main algorithms were implemented and used for the recommendation process. Support vector classifier to classify documents to their classes based on schools or philosophies. Item content computation (item-to-item content similarity) also computed using tf-idf. And a k -Nearest Neighbor algorithm was implemented for finding the k most similar users to the active user and finding recommendations based on users' rating history. Finally, a hybrid approach between these two has been implemented by combining the results given by each of them.

(ii). Proposed Framework

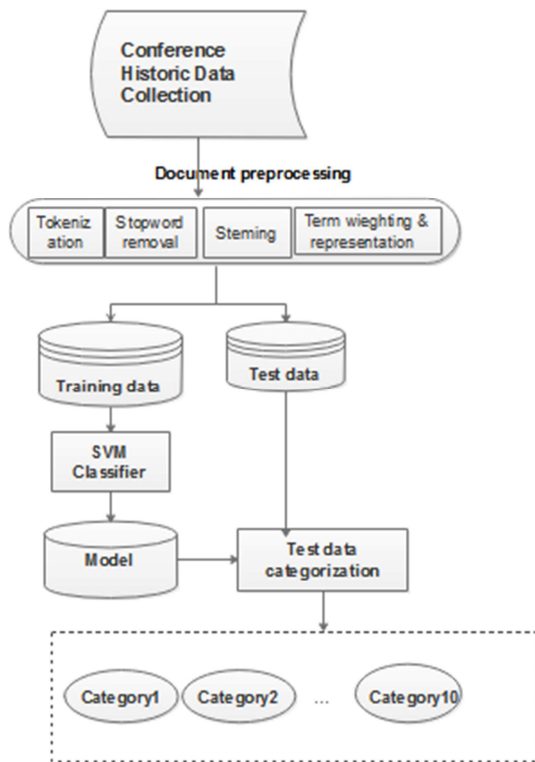


Figure 2. Classifier Modeling.

This study proposed a hybrid personalized research conference presentation papers recommendation, which is a framework based on classified reviewed papers that is able to

select papers set and generate recommendations that fit the user profiles. It is aimed at delivering relevant domain specific research call services to conference participants with the least user input. As new paper articles enter the stream of paper articles collection, it is classified based on its content and metadata. At the same time, it also constructs the user profiles based on their specific domain study, paper ratings and similar users.

This study's framework is based on Classification Technology techniques such as TFIDF/Term Frequency Inverse Document/weighting scheme and SVM/support vector machine classifier. Data preparation, classification modeling, training and testing data are the three stages. Modeling the classifier consists of two phases: (1) model training, which occurs during the system designing and building process, as shown in figure 2; and (2) model applying, which is an interactive part that is directly connected with the participant interface.

The model training phase addresses the following issues: creating a participants' feature space, creating a paper article label space, and training a classifier to deal with the relationship between them. Using the model's output features as input vectors to train the data. The trained data is then used to classify new data. The SVM classifier model's resultant classified data is then sent for further recommendation using both collaborative filtering and content-based techniques.

The proposed framework, as depicted in the figure 3 is comprised of six main components: User Interface, Schedule Management, Dataset component (Users profile store and processed papers Data Store), Content-Based filtering, Collaborative filtering and Recommendation Engine components.

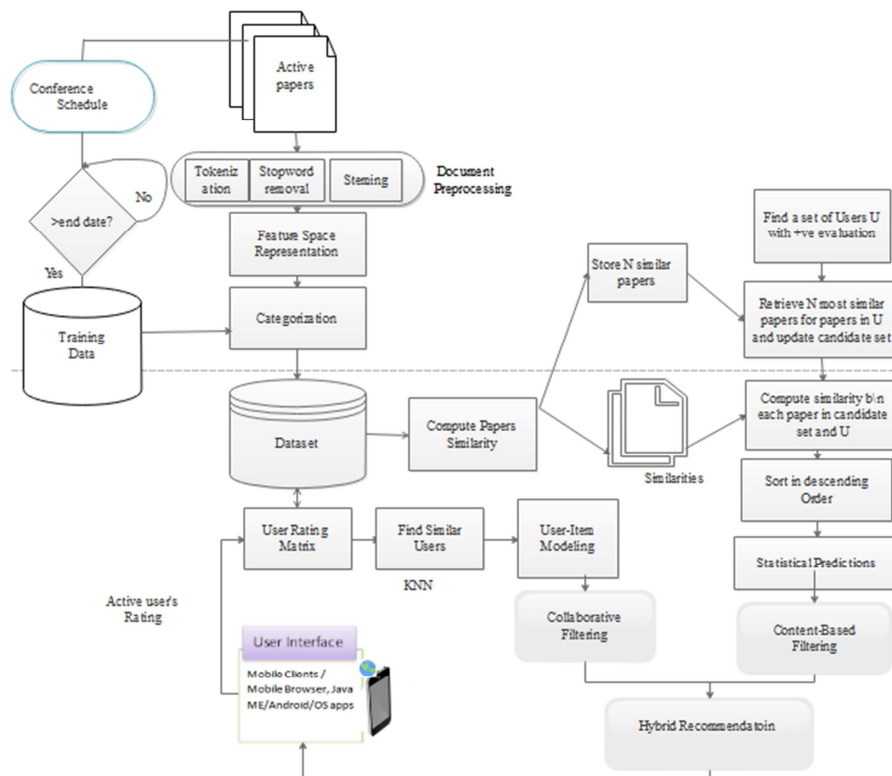


Figure 3. Proposed Research Recommendation Framework.

The framework finds the similarity between papers' content and combine collaborative filtering based on user ratings and finally return a set of papers that meets the user interest and preferences on their smart phone based on research conference schedules. The nobility of this framework that makes differ from others is the delivery of items for rating that are presented to the user is based on a measure employed by SMO classifier and hybridization of content based and collaborative filtering techniques to eliminate individual defects.

The major issue the researchers employed classifier is because every participant has only one school. For instance, in movie recommendation, a user may be more likely want to watch comedy, which doesn't mean he/she dislike other movies with different genres. For the researcher knowledge, there is no recommendation framework which identifies categories between items to be rated by users. A user with health research background shouldn't be asked to rate engineering research articles. These issues did the conceptual design impossible to collect user ratings based on item popularity. Additionally, the framework eliminates Sparsity and cold start problems by combining content based filtering and collaborative filtering. It also includes time context and user profile modeling, which is not included in paper recommendation literature.

(iii). User Interface Component

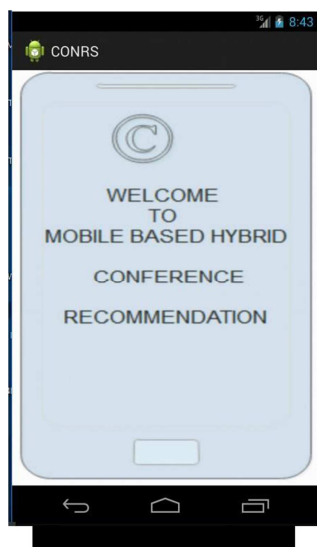


Figure 4. First page of the proposed system.

The Interface Layer of the research conference papers recommendation connects the recommender system's user to the computation layer. It facilitates interaction between users and the system of a service provider. The user interface is the visible portion of the interface layer, through which users can request and view recommendations. The more functional portion of the interface layer, on the other hand, is hidden from the end user [62, 34]. This section validates and translates the ephemeral and persistent user requirements, communicates this information to the computation module,

and formats the returned recommendation list to provide the user with useful recommendations. User interaction occurs through user registration, login, and corresponding recommendations. It serves as an interface between the system and active users. It will be generated based on the information provided by the respective user, including personal information and interests within a given domain of research conference papers.

As shown in Figure 4, the user first interacts with the "Welcome Home" page of the system. When the Welcome home page touched or clicked, the system displays another page (figure 5). On the main page, there are two main parts: Login and Register buttons.

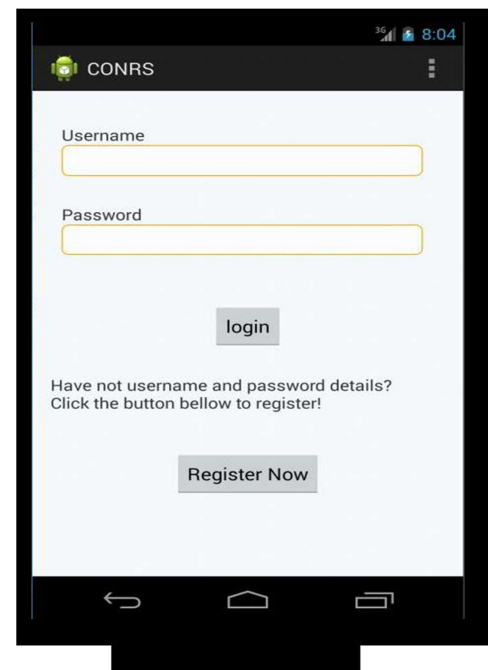


Figure 5. User login and Registration Page.

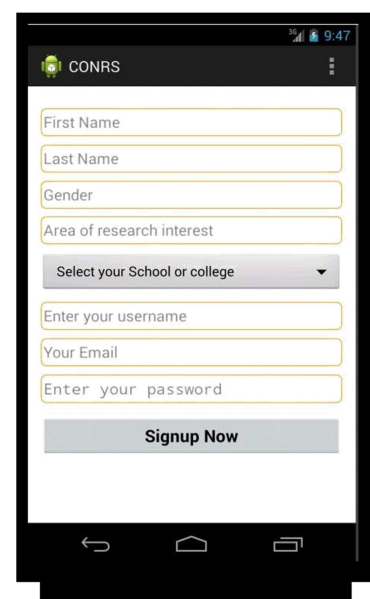


Figure 6. User Registration Page.

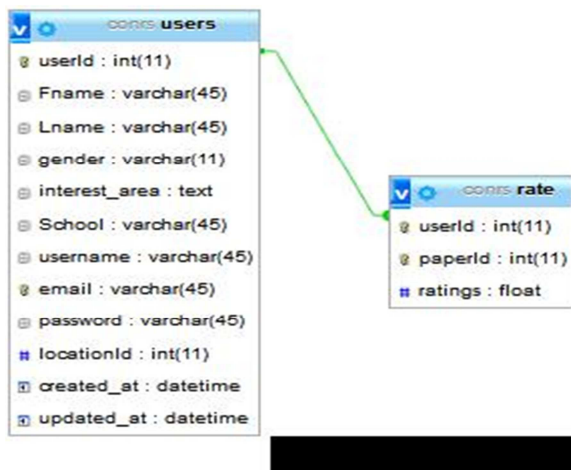


Figure 7. User Item Rating.

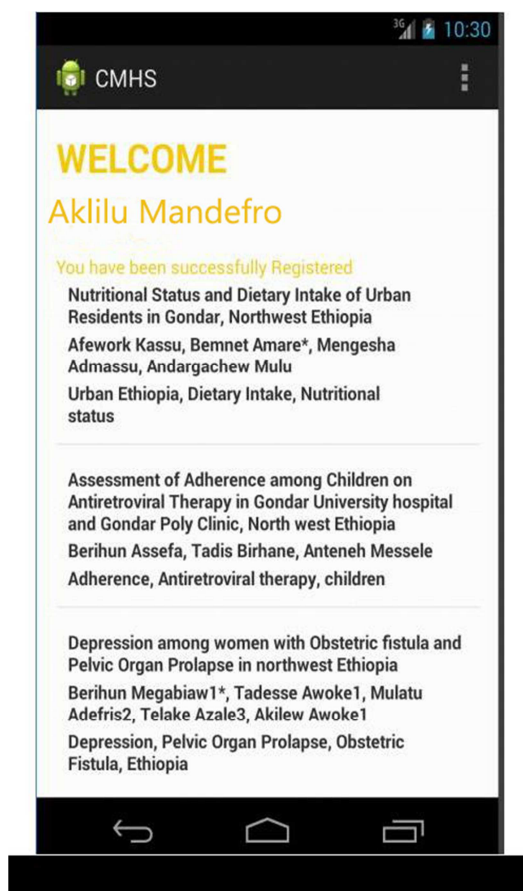


Figure 8. Provided Articles Based On User Registration Information.

When the “Register” button is clicked, the register user interface appears. As shown in figure 6, the user first registered in the system in order to access the service. When the user has correctly filled all the required information properly and submitted to the system, the recommendation engine crosschecks all the required information through the web service and then stored in the database. After the required fields are provided, the recommendation engine responds back a “User successfully registered” message is displayed from the web service through Json Parser to a mobile enabled user interface. After the user registered, the

system provides the categorized articles based on the user’s school as depicted in figure 8. Then they have to go provide ratings for the articles they are interested. Based on the feedback the user given the profile will be created and based on similar rating history and similar articles of most liked papers depicted in figure 9 recommendations is provided.

After the user registered with necessary demographic information and interest the user finds the articles categorized based on his/her School or College. As depicted in figure 8, the registered user is choosing Health and Medical Science School. The interest area is matched Based on it and he/she provided those articles fetched by the Json parser from the database. Then the user can rate the articles displayed based on 1-5 Likert rating scale. Those rating information about the user and article information are stored in the *rate* table of the CONRS database as shown in figure 7. Here it is the figure for rate and users table:

Since we use List android activity, users can slide down and up to read articles and they could give rate for items they like so that they can be recommended similar articles using CBF and most rated articles rated by other like-minded users.

Recommendation Scenario

Here the above user rated 10 research articles the most he/she likes. Based on rating information the CBF component selects one most rated article and find most three similar articles as candidates, and CF component finds most five candidate articles from like-minded users. Finally the system hybridized using a union function to final recommendation as shown below in figure 9.



Figure 9. Recommended Articles for the Scenario.

(iv). Schedule Management Component

Several lines of research have successfully used multi-criteria ratings to improve recommendation accuracy. Context Aware Recommender Systems were created after Adomavicius et al. proposed a contextualized view on ratings [1]. The concept of context-awareness corresponds to the pervasiveness of mobile devices. Mobility introduces several contextual dimensions that are either implicitly fed (e.g., change of location) or inferred (e.g., multiple visits or spending more time than average in a POI/point of interest/ may be regarded as a positive 'vote') [43]. In this study, the component is a server-side time stamp component that adds date/time stamps to data that is supposed to be persistently stored in the provider's repository. The time stamp in a provider's schedule contains the start/end time as well as the date on which an event/activity occurs at a specific location. This component assists us in incorporating historical data into the classifier model.

(v). Storage Component

(a). User Profile Store

To derive content recommendations, a significant number of mobile RSs/recommender systems/ rely on user constraints and preferences, either explicitly stated or implicitly inferred [48]. The explicit user profile is typically created during application startup via a brief survey that includes demographic information, constraints, preferences, and user goals. As the user interacts with the system, the implicit user profile is fed, implicitly denoting preference for

certain items (via interaction behavior/history, ratings and critiques on recommended items). The User Data Collector component is in charge of gathering user information (such as demographic information, personal interests, and domain-specific research preferences) as well as the users' research conference service attendance history. Figure 10 shows how users with similar profile information have been categorized into one group. The information gathered is passed on to the recommendation engine component, which generates appropriate recommendations.

(b). Paper Data Store

The papers data store is in charge of storing all papers and research conference caller data needed to generate personalized research conference services. Document classification is the process of organizing similar documents into groups, which is essential for document organization and retrieval [48]. Papers in scientific paper archives such as Google Scholar come from a variety of disciplines, including math, biology, computer science, and economics. Each group has a different set of topics. Computer science papers, for example, cover topics such as operating systems, networks, machine learning, and so on, whereas economics papers cover topics such as entrepreneurial economics, financial economics, and mathematical economics [51]. However, in this study, researchers categorize documents based on their school and philosophy for the Sparsity problem, and participants are also classified based on their school of study.

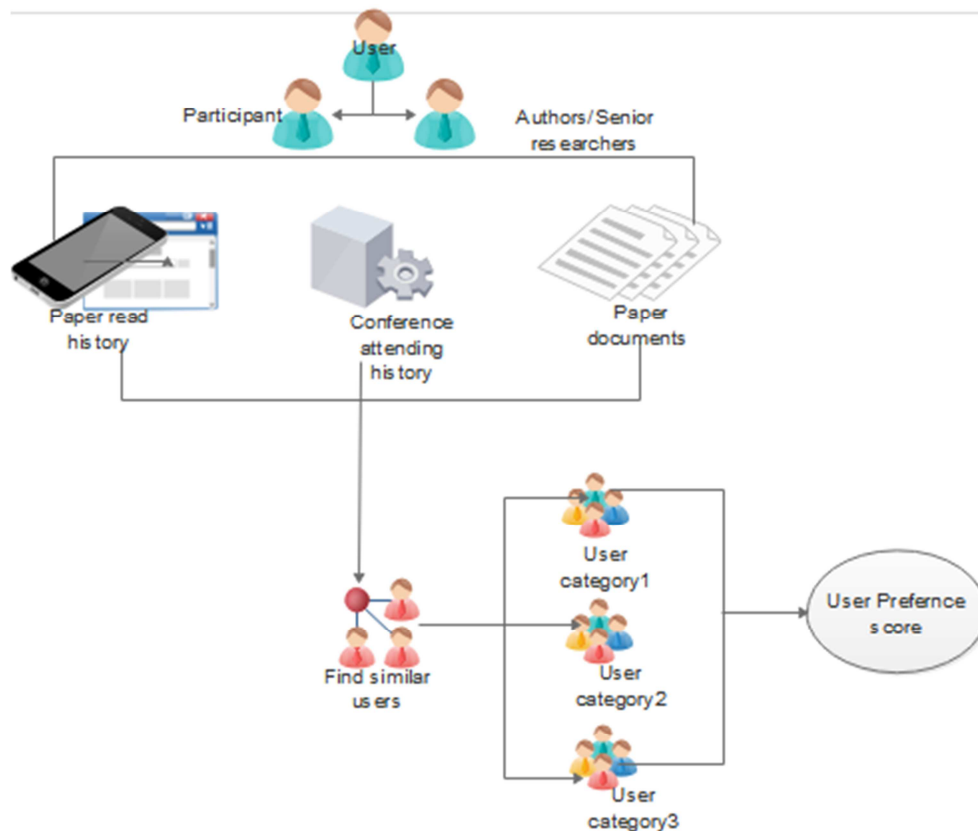


Figure 10. Workflow diagram for categorizing similar users.

(vi). The Content-Based Filtering Module

Article features in content-based prediction include title, abstract, and keywords. The content-based module determines the user's research preferences by analyzing the article content [41, 57]. First, this unit analyzes user logs and retrieves research conference article content. Second, the retrieved contents are processed at the term level, with terms extracted and split. Third, the papers' TF-IDF/term frequency - inverse document frequency/ vectors build on previous processes' results. Fourth, the SVM classifier produces classification at the term level as well as prediction at the term level. Following that, this module discovers the relationships between the predicted results of papers and the user's preferences (as a query) to provide the content-based module's final decision.

(vii). The Collaborative Filtering Module

The researchers used an item-based collaborative filtering module to find like-minded readers of the current user and make recommendations to improve recommendation efficiency. The operation of this module consists of two steps: (i) locating like-minded readers who have the same rating patterns as the active users, and (ii) predicting the active user's interest in the new item based on the rated item similarity of the like-minded users.

(viii). Recommendation Engine Component

The recommendation engine module employs a linear model to combine the content-based and collaborative filtering modules' decisions into a final decision. The Recommendation Engine component is in charge of making recommendations to users about the most relevant services based on their interests and/or requests [33]. The Recommendation Engine component computes user-paper preference based on the Content and User Similarity (KNN) algorithm [26, 67]. Content-based algorithms typically compare a representation of the user profile to (the metadata of) the content and recommend the top-N items. As a result, it detects similarities between conference papers and user preferences, as well as similarities between rating profiles, and makes predictions based on this data. Figure 11 depicts how the output of the two components is combined using a linear weighted scheme with the union or intersection functions.

When the client application receives a request from the interface for new research conference papers, the request is routed to the server application. Based on the data provided to recommendation engines, a set of research conference papers is then recommended.

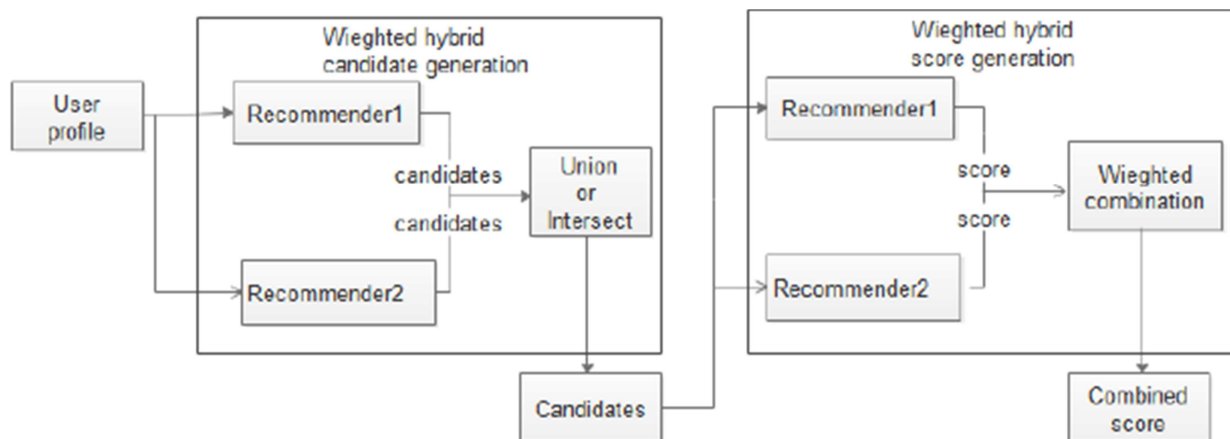


Figure 11. Weighted hybrid technique.

2.6. Data Description and Experimental Setup

This study uses unlabeled text documents collected from different sources for categorizing research paper documents that helps as background data for the system. The researchers created data sets using unstructured data collections found in university of Gondar and Bahirdar university research centers. The first step is categorizing papers to classify to their class as described in table 4, then use those categorized data as input together with user information for the design of conference recommendation framework.

The research conference papers recommendation has (i) papers as background data, the information that the system

has before the recommendation process begins, (ii) input data, the information that the user must communicate to the system in order to generate a recommendation, and (iii) an algorithm that combines background and input data to arrive at its suggestions.

To test the performance of the framework, the researcher used 271 paper articles collected from University of Gondar and Bahirdar University presented at annual research conferences. A multi-disciplinary recommender system to advise research resources in University Digital Libraries used a classification composed of 25 disciplines to represent resources [16]. This study summarizes disciplines into 10 categories as listed below in table 4.

Table 4. Resource Scope.

№	Research Domain	Number of documents		Total	Removed
		2013	2016		
1	MEDICINES AND HEALTH SCIENCES	45	37	82	
2	BUSINESS AND ECONOMICS	8	22	30	
3	NATURAL AND COMPUTATIONAL SCIENCES	9	21	30	
4	SOCIAL SCIENCES AND HUMANITIES	13	23	35	1
5	AGRICULTURE	11	7	18	
6	VETERINARY MEDICINE	13	23	36	
7	TECHNOLOGY	1	4	5	
8	EDUCATION	7	8	15	
9	LAW	3	6	9	
10	SCHOOL OF INFORMATICS	0	11	11	
Total		110	162	271	

The goal of this section is to classify the given specified experimental dataset into ten categories. The paper articles cover Medicines and Health Sciences, Business and Economics, Natural and Computational Sciences, Social Sciences and Humanities, Agriculture, Veterinary Medicine, Technology, Education, Informatics and Law. Researches done with Amharic language is removed from the collection.

2.6.1. Phase 1: Preprocessing

Documents are represented by feature vectors in this phase, which are then used to create a training data set for the machine learning phase. This conversion demonstrates the fundamental concepts of the vector space document model, which is critical in information retrieval and web search [22]. To achieve the goals of this phase, various approaches and tools can be used. However, as mentioned in the preceding section, the Weka machine learning system library is used [52, 55].

a) Creating a String Data File in ARFF Format

To accomplish this step, first concatenate all text documents (text corpus) obtained from the data collection step and save them in a single text file, where each document is represented on a separate line in plain text format with its main attributes and class, by loading all text files in MS Word and then saving the file in plain text format without line breaks. Although the WEKA SimpleCLI feature can automatically convert the text document collection into a Weka readable format, we must manually label the document with the necessary features for later use in the recommendation system.

Once the text corpus file is created, each line (individual document content) must be enclosed in quotation marks ("), a document name or ID must be added at the beginning of the line, and the document topic (class) must be added at the end, all separated by commas. A file header is also required at the beginning of the file, followed by @data, as shown below:

```
@relation conference_papers
@attribute paper_id string
@attribute paper_title string
@attribute paper_author string
@attribute paper_abstract string
@attribute paper_keyword string
@attribute paper_class string
@data
```

"CMHSA1", "Prevalence of Mortality and Associated ...", "Abera Shibrul, Berihun...", "Mother to child transmission of HIV/AIDS during pregnancy, delivery, breast feeding and...", "Mortality, HIV-infected, Weaning, North Gondar", CMHSA.

"CMHSA2", "Rapid Diagnosis of Tuberculous Pleuritis and Lymphadenitis with ...", "Agerie ...", "A rapid, sensitive and accurate laboratory diagnosis has paramount importance in cases of suspected ...", "immunocytochemical pathology laboratory", CMHSA.

"FBEA1", "Assessment of the Performance of Ethiopian Financial Institutions", "Abebaw Kassie...", "The financial system plays a pivotal role in economic activities in any country...", "FBEA.

b) Tokenization

Tokenization is one of the preprocessing tasks [24]. WEKA implements the tokenization of corpus documents and user-supplied input strings. After the document features have been processed and tokenized, the system extracts the document's nouns and verbs from the token set in order to preserve the semantics of the document. Three major activities were performed on the token set's semantics set: stemming, stop word removal, and normalization [50].

c) Stop Word Removal

In this study, researchers extracted the paper features from publicly available portions of documents. Those parts of documents are title, keyword and abstract. The researchers implemented stop word removal for those document features using a prepared list of English language. It is common to find that several attributes are useless (such as the word "a", "the", etc.). Thus, stop word removing algorithm has been applied to words from a file. To initialize the algorithm a set of stop words has set by the human beforehand and hence stored in a text file. Then, the model can simply match the attributes with those preset stop words [36].

d) Stemming

The snowball stemming algorithm is the third algorithm used in the preprocessing phase. Because some words have similar meanings but different grammatical forms (for example, "research" and "researches"), they must be combined into one attribute. As a result, the documents will have a better representation (with stronger correlations) of these terms, and the dataset will be smaller, resulting in a faster processing time. At this stage, irrelevant terms are

removed from the documents, and words from the same context but with different forms are combined to form the same word. This stage's final goal is to convert the text collection into a matrix of index terms with their tfidf weight values.

2.6.2. Phase 2: Feature Selection

One of the most important preprocessing steps in data classification is feature selection [42, 59]. To remove noise features, it is an effective dimensionality reduction technique. The basic idea behind a feature selection algorithm is to search through all possible attribute combinations in the data to determine which subset of features works best for prediction. Thus, the number of attribute vectors can be

$$I(w) = \sum_{i=1}^k p_i \log(p_i) + F(w) \sum_{i=1}^k p_i(w) \log(p_i(w)) + (1-F(w)) \sum_{i=1}^k (1 - p_i(w)) \log(1 - p_i(w)) \quad (5)$$

The greater the value of the information gain $I(w)$, the greater the discriminatory power of the word w . For a document corpus containing n documents and d words, the complexity of the information gain computation is $O(n.d.k)$. The following figure summarizes the feature selected by applying information gain or entropy evaluator.

Attribute Evaluator (supervised, Class (nominal): 1354 @@class@@):
Information Gain Ranking Filter

Ranked attributes:

0.75	1123	cncs
0.75	260	tessema
0.75	209	regression
0.75	85	engagement
0.75	286	wondimnew
0.75	78	economics
0.75	44	cbe
0.75	112	haimanot
0.75	104	gedif
0.75	140	lecturer
0.75	426	ethiopia
0.75	61	correlation
0.75	111	guadie
0.75	97	fekadie
0.744	1066	using
0.708	629	sectional
0.708	380	cross
0.666	263	three
0.655	150	main

Figure 12. Sample Features selected in rank.

3. Evaluation and Testing

a. Processed Data description

To build and evaluate the classification model, the total

reduced by keeping the most meaningful ones and removing or deleting the irrelevant or redundant ones [56].

In this study, all of the documents in the training data are classified into ten distinct categories, from which the model can simply compute which terms are frequently occurring in each. Some useless or irrelevant attributes can thus be filtered out. A gain or entropy evaluator is used to obtain the final feature set information. Let P_i is the global probability of class I , and $p_i(w)$ is the probability of class I , given that the document contains the word w . Let $F(w)$ be the fraction of the documents containing the word w .

The information gain measure: $I(w)$, for a given word w is defined as follows:

271 documents has divided into two datasets, namely training set and testing set, in which 70% of the documents used to the training set, whereas the remaining 30% used for the testing set.

In the representation of these documents, 271 instances have been vectored into 6728 attributes (in term of numerical values). No missing data are among the attributes and all the numeric attributes are described in the term frequency/inverse document frequency (TFIDF). The data presented in Figure 13 and Table 5 summarizes the descriptive data in both training and testing set.

Terms that were in the documents or in the query needed to get hold of weight with respect to their documents are selected based on attribute selection method [67]. The weight can be calculated using *tf-idf* term weighting methods. The term weighting was done by finding the frequency of the terms and its synonyms in the documents and product of its inverse document frequency. The most important thing was once weight of a term was calculated, its synonymy term weight never been calculated. because its frequency already adds up to the coming synonymy term.

Table 5. Data Description table.

	Training data	Test data
Number of instances	186	77
Number of attributes	6734(numeric-6733, nominal-1)	6734(numeric-6733, nominal-1)
Missing data	No	No

telation: C:\Users\Nigusie\Desktop\T-weka.filters.unsupervised.attribute.StringToNominal-R1ast-weka.filters.unsupervised.attribute.StringToNominal-R1ast-weka.filters.unsupervised.attri

No.	1: able	2: absorption	3: abstract	4: accounted	5: achieve	6: adapted	7: administer	8: affect	9: affects	10: also	11: always	12: amhara	13: among	14: ample	15: analysing	16: analysis
	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric
1	1.82...	0.0	0.0	1.829255...	0.0	1.8292...	1.829255...	0.0	0.0	0.0	0.0	1.34880...	0.0	0.0	0.0	0.0306255...
2	0.0	0.0	0.0	0.0	1.0677...	0.0	0.0	0.0	0.0	0.48...	0.0	0.0	0.0	0.0	0.0	0.0306255...
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.86834...	0.0	0.0	0.0306255...
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	1.829255...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48...	1.8292...	0.0	0.86834...	1.8292...	1.8292551...	0.0306255...
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48...	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48...	0.0	0.0	0.0	0.0	0.0	0.0306255...
9	0.0	0.0	0.0	0.0	1.0677...	0.0	0.0	1.82...	1.829...	0.0	0.0	0.0	0.0	0.0	0.0	0.0306255...
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48...	0.0	1.34880...	0.0	0.0	0.0	0.0306255...
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.86834...	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0306255...
13	0.0	0.0	0.0	0.0	1.0677...	0.0	0.0	0.0	0.0	0.48...	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.48...	0.0	0.0	0.86834...	0.0	0.0	0.0306255...

Figure 13. TFIDF vector representation.

b. Testing the Performance of the Prototype

Due to the intrinsic features of recommender systems, it is difficult to apply statistical-method to compare different recommender frameworks with each other [20, 34]. The developed prototype for the framework is tested and evaluated to check the objectives of the research are achieved. Performance testing is the process of determining the speed or effectiveness of a developed prototype system. The performance of the system in this study is tested by using a confusion matrix test. A confusion matrix contains information about the actual and predicated classifications done by experts and recommendation system. In the context of information retrieval, the most widely used methods of confusion matrix in this study are precision, recall and F-measure. Precision is the fraction of recommended documents that are relevant to the test user, while recall is the fraction of the documents that are relevant to the user which are successfully recommended [6, 23]. A single measure that trades off precision versus recall is the F-measure, which is the weighted harmonic mean of precision and recall [43].

The performance of the prototype is evaluated using ten classes and ten users to test. In each field of study five users rate at least 10 papers for profile learning purpose. To evaluate the performance of the system ten purposive users in each domain are selected from registered users, one from Informatics, the second from Business Economics, the third from Medicine and Health Science and so on for each field domain. The prediction of the performance of this prototype system is evaluated by using the data in the confusion matrix given in Table 6.

It should be noted that the testing dataset contains only 271 scientific articles written in English with the following distribution: 80 articles belong to the Health category, 30 to the Business and Economics, 35 to the Social and Humanity, 30 to the Natural Science category, 36 to the Veterinary Medicine, 18 to Agriculture, 5 to the Technology, 15 to Education, 9 to Law and 11 to Informatics category. The experiments in this study did not aim to measure the performance of the strategy used in terms of execution time, but only the relevance of the recommended articles to the input article.

Table 6. Shows the confusion matrix of a prototype for testing the performance of the system.

CONRS predication												
	Test cases	CMHS	FBE	NCS	SSH	FVM	SoL	SoT	SoE	FoA	INFO	Total
Actual data from users as experts	CMHS	24	0	0	0	0	0	0	0	0	0	24
	FBE	1	7	0	1	0	0	0	0	0	0	9
	NCS	0	0	9	0	0	0	0	0	0	0	9
	SSH	0	1	2	7	0	0	0	0	0	0	10
	FVM	1	0	0	0	8	0	0	0	1	0	10
	SoL	1	0	0	0	0	1	0	0	0	0	2
	SoT	0	0	0	1	0	0	0	0	0	0	1
	SoE	0	0	0	1	0	0	0	3	0	0	4
	FoA	0	0	0	0	0	0	0	0	5	0	5
	INFO	0	0	0	1	0	0	0	0	0	2	3
Total		27	8	11	11	8	1	0	3	6	2	77

We used standard evaluation measures of information retrieval systems: precision, recall and F-measure (a combination of precision and recall) [16], defined in equations (5), (6), and (7).

$$\text{Precision} = \frac{\#(\text{good items recommended})}{\#(\text{all items recommended})} = p \quad (5)$$

$$\text{Recall} = \frac{\#(\text{good items recommended})}{\#(\text{total number of full items})} = r \quad (6)$$

$$\text{F-Measure} = \frac{2pr}{p+r} \quad (7)$$

Based on the table 6 and equations 5, 6, and 7, we get the following result table for precision, recall and f-measure.

Table 7. Precision, recall and f-measure table.

	Precision	Recall	f-measure
CMHS	0.89	1.00	0.941239
FBE	0.88	0.78	0.823654
NCS	0.82	1.00	0.89989
SSH	0.64	0.70	0.666467

	Precision	Recall	f-measure
FVMA	1.00	0.80	0.888889
SoLA	1.00	0.50	0.666667
SoTA	0.00	0.00	0.00
SoEA	1.00	0.75	0.857143
FoAA	0.83	1.00	0.908893
INFO	1.00	0.67	0.80024
Average	0.858	0.857	0.849

As it is stated in the previous sections, the researchers set train data to 70 percent and the test data to 30 percent to test recommender framework prototype. In the above precision, recall confusion matrix table, we can observe that the system recommends 27 articles from 24 articles given as a test for CMHS. The system recommends 3 other articles from other fields 1 from Veterinary medicine articles, 1 from Business Economics and 1 from Law articles. Therefore the precision of the test in this case $24/27 \times 100\% = 0.89$, is the proportion of positive predictions that are correct (no. of good papers recommended / no. of all recommendations) which is 89 % and the recall which is the true positive rate defined as no. of

good papers recommended / no. of good papers 24/ 24*100%, which is 100% of a prototype system.

For test category FBE, the predication of the system among 9 articles, 7 are correctly predicted and 2 others are recommended for Health Science and Social Science member user. In the test case NCS the predication of the system among 9 test articles, 9 are recommended. For the test case FVM, 8 articles are correctly recommended to the intended user and the other 2 are recommended for CMHS and FOA users which are the effect of content-based filtering.

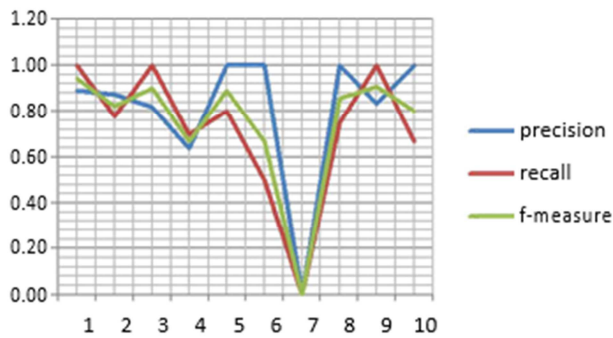


Figure 14. Precision Recall and F – Measure.

According to Table 7 above, the performance of a prototype system shows correctly recommended test case articles by the domain users are totally 66 out of 77 cases. Which is $66/77 = 0.86 * 100\% = 86\%$. Therefore, the performance of a prototype system in this study based on domain users registered 86%. The rest 14% lacks the performance of the system because of machine learning defects. Based on the articles used for testing for each test class in the system as indicated in Table 7, the following graph (figure 14) shows summarized precision, recall and f-measure below.

During experimentation, the researcher observes that the recommendation performance increases as the training data increases, which were tested in each category. Categories such as CMHS, FBE, NCS, SSH and FVM have more data than others in the training dataset which also obtain a better F-measure values and the reverse minimizes the performance of the system.

4. Experimental Results and Discussion

4.1. Experimental Results

The experiment compares the performance with four commonly used learning methods for text categorization in our dataset. Each method represents a different machine learning approach: density estimation using a naive Bayes classifier, a distance weighted k -nearest neighbor classifier with IBK algorithm, and the J48 decision tree. SVM training is carried out by the SMO/sequential minimal optimization/algorithm. The objective of this evaluation is twofold. We have created two datasets using the described data; Binary/Boolean and TFIDF datasets. First, it compares the classification accuracy and performance when different classifiers are applied to Binary dataset. Second, it compares the classification accuracy and performance when different classifiers are applied to TFIDF dataset.

A dataset with 271 documents classified in ten different categories is used for evaluation. The selected dataset contains categories of document: CMHS, category for Health Science papers; FBE, category for Business and Economics papers; NCS, category for Natural and Computational Science papers; SSH, category for Social Science and Humanity papers; FoAA, category for Agriculture papers; VMA, category for Veterinary Medicine papers; SoTA, category for Engineering and Technology papers; SoEA, category for Education papers; SoLA, category for Law papers and INFO, category for Computer and IT related papers. 70% of data (i.e. 186 documents) are extracted randomly to build the training dataset for the classifier. The other 30% documents extracted randomly from the full data set are used as the testing dataset to test the classifier. The classification task considered here is to assign the documents to one or multiple categories of the 10 research paper categories. A document belongs to a category if it is indexed with at least one indexing term from that category. Table 8 shows the results on the binary dataset of the four text classification algorithms. The performance of the four algorithms with binary dataset presented.

Table 8. Performance of algorithms with Binary dataset.

Binary dataset						
No	Algorithms	Correctly classified Instances	Incorrectly classified Instances	Precision	Recall	F-measure
1	Naïve Bayes	67.53%	32.47%	0.228	0.325	0.173
2	SMO	83.12%	16.80%	0.844	0.831	0.819
3	IBK	75.32%	24.68%	0.83	0.753	0.744
4	J48	72.73%	27.27%	0.772	0.727	0.716

As the results shown in table 8, SMO correctly classify 83.12%, which is 64 research papers on the test set and incorrectly classify 16.88% of test dataset which is 13 research papers. Naïve Bayes classifies 67.53% correctly and

32.47% incorrectly. The IBK algorithm correctly classifies 75.32% and incorrectly classify 24.68% of the supplied dataset, and J48 classifies 72.73% correctly and 27.27% incorrectly in binary dataset.

Table 9. Performance of algorithms with TFIDF dataset.

TFIDF dataset						
No	Algorithms	Correctly Classified Instances	Incorrectly Classified Instances	Precision	Recall	F-measure
1	Naïve Bayes	81.82%	18.18%	0.82	0.818	0.806
2	SMO	83.12%	16.88%	0.842	0.831	0.817
3	IBK	79.22%	20.78%	0.844	0.792	0.782
4	J48	75.32%	24.68%	0.755	0.753	0.739

The above table, table 9 summarizes the result of using SMO correctly classify 83.12%, which is 64 research papers on the test set and incorrectly classify 16.88% of test dataset which is 13 research papers. Naïve Bayes classifies 81.82% correctly and 18.18% incorrectly. The others, IBK and J48 perform less than Bayes and SMO classifiers in a dataset prepared TFIDF vector space as their classification accuracy shows in experiment result. In prior studies describes why SVMs algorithm should work well for text categorization [63, 64]. It uses over fitting protection, which does not necessarily depend on the number of features; they have the potential to handle large feature spaces (more than 10000). The results in table 8 and 9 shows that features even ranked lowest still contain considerable information and are somewhat relevant. A classifier using only those worst features has a performance much better than random. Since it seems unlikely that all those features are completely redundant, this leads to the guesswork that a good classifier should combine many features and that aggressive feature selection may result in a loss of information. That is why SMO perform equally in both Boolean and tf-idf weighting schemes.

4.2. Discussion

A mobile based framework for conference paper recommendation system is implemented using eclipse (Java) android programming language, PHP scripting language and Weka machine learning tool. The experimental results of the developed system for the designed framework are shown in Figure 12 up to Figure 14. The researcher has taken different test cases and analyzes the data by 77 articles and 10 member users. The response obtained from domain users as the expert has got very good acceptance and they appreciated the idea. Based on domain users, the predication of the performance of a prototype system is evaluated by using the data in the confusion matrix. The recall or completeness of the system performance for CMHS category is 100% and the precision or exactness of the system performance for the same category is 89%. In all sum average performance, the system has shown good accuracy when performing the task which is 86%. The rest of defects in the system indicate the predication of test case which is 1 article in case test FBE suggested as CMHS, 1 as SSH and also 1 FVM article recommended as CMHS, 1 as FoA. This indicates, there is a rare test case incorrectly classified by the system.

Additionally, the researcher used seven questioners to end user and domain users for the purpose of evaluating the user acceptance of a prototype system. Eleven system end users are selected randomly and purposively to provide a value for

each question. The different scaling rate, which is given as poor = 1, fair = 2, good = 3, very good = 4 and excellent = 5. Based on these criteria, none of the evaluators responded the system as poor and fair. Nine of responses are obtained from evaluators as good, 33 responses as very good and 31 responses as excellent. Due to this, the researcher found satisfactory results and calculated user acceptance based on users' comment is 86.5%. Generally, this prototype has got acceptance by domain users and end users in research conference paper recommendation system.

5. Concluding Remark and Recommendations

Mobile based systems are not widely used in our country, Ethiopia in the area of recommender systems field, especially for the digital library resources, and conference item recommendations. In this study, mobile based system is developed for a conference papers recommendation which is going to be available for the near future of research conferences.

5.1. Concluding Remark

Research call services must provide information proactively and tailor their services to users. One way to make RS/recommender system/ development more independent of software developers, and consequentially reduce cost and development time is to create a framework for automatic RS generation. The first step in producing this framework is to represent information about users (participants) and items (papers for research call presentation).

In this research work, a framework for mobile based conference paper recommendation and its prototype implementation with researchers' friendly language was presented. Researchers have been conducted a literature review, understood the conceptual framework of recommendation systems and selected a hybrid recommendation approaches which tries to use the advantages of one to fix the disadvantages of other techniques. The prototype implemented using Hybrid approach with three different algorithms, Content based filtering algorithm, a k-Nearest Neighbor algorithm (CF), and Classifier technology using Weka tool. Mainly the recommendation is designed based on a linear model which combines the recommended scores of the content-based module and the collaborative filtering module. After designing the conceptual framework, the researchers implemented the prototype in android based emulator using

eclipse Java and PHP programming language.

In testing and evaluation of a prototype system, 10 domain experts are selected by purposive sampling technique for each test category. The recommended articles are identified by comparing decisions made by the domain experts and recommender system based on different test cases of classes. The process ensures that the prototype satisfies the requirements of its end-user and the result shows that the system registered 86.5% of the user acceptance. The overall performance of the prototype system result is 86%. Generally, the prototype achieved a good performance and met the objective of the study.

Although our experiments did not aim to measure the performance of the strategy used in terms of execution time, it measures the relevance of the recommended articles to the input article. The challenge observed in the developed system is that when the number of papers increased the content-based module takes long execution time to be displayed on the android virtual device (AVD). The researcher is not trying to solve it due to time constraints.

On the other hand, the majority of users are interested in the implementation of the application system being applied. The attitude of participants towards utilization of mobile phone for research conference service is very interesting. With the study of this research, it is expected application developers can do more about mobile based research conferences to allow users meet the information needs of their research ideas.

5.2. Recommendation

No research organization ever announced papers going to be presented on their conference. Because of these participants have never known what research idea is presented and examined. To resolve such problems, the researchers designed this framework and developed a mobile based conference recommendation system to support participants. The researchers recommended that in the future such type of systems will be helpable for conference providers and participants so as to be implemented. Based on the findings of the study, the following recommendations are furthered as future works and research opportunities for practice in the domain area.

6. Future Works

To the application target, the future work of this study is to deploy the proposed framework as the backbone of a research conference recommender system in a mobile environment.

- 1) Case studies in each field of study are needed for Local ontology development to eliminate articles which are classified in false classes using cases or rules to construct participating with each school field expert. This can help to integrate ontologies and/or encyclopedic knowledge.
- 2) During this framework development, we manually collect research articles from two university research

Centers: University of Gondar and Bahirdar University. We faced incomplete information about services and articles. For current research problem, none of them use a system for their conference issues available for presentation held on some days. To implement and deploy the system in real environment, the research conference providers should prepare a system that record conference services.

- 3) For simplicity purpose, researchers used single criteria user rating matrix (Likert-scale) in prototype development. In the future, it is better to incorporate multidimensional ratings, which can include research properties identified in literature review such as current magnitude of the problem, scientific opportunity, capacity strengthening, direct implication for policy development and others.
- 4) Furthermore, the significant explosion of research papers in conferences has made it difficult for researchers to easily access relevant scholarly papers for academic learning. Conferences, in comparison with journals, have an aspect of social learning and networking, which leads to personal familiarization through various interactions among researchers. So, it is better to include the social awareness of research conference participants by computing the social ties between other conference participants based on their papers interest and participation history.

7. Contribution

The contribution of this study is to initiate future researchers in which they can do more in the area of mobile based recommendation systems. In addition, this study will be an input to further studies for smart research conference and it is a problem solving throughout the country when it is implemented by research practitioner organizations. Around three hundred (300) papers prepared for machine learning purpose in binary arff file format and tfidf arff file format which can help students in university laboratories, digital libraries and further research study with more articles included.

References

- [1] G. Adomavicius et al. "Context-Aware Recommender Systems. *Association for the Advancement of Artificial Intelligence*", 14. 2011.
- [2] Wolfgang woerndl, christian schueller and rolf wojtech, "A hybrid recommender system for context-aware recommendations of mobile applications", data engineering workshop, 2007 ieee 23rd international conference, pp 871 – 878.
- [3] Perusco, L. (2005) Human centric Applications of Precise Location Based Services, Bachelor of Information and Communication Technology (Honours), University of Wollongong, 98p.

- [4] Marla, et al (2011). "Studies Orientation and Recommendation System (SORS): Use Case Model and Requirements".
- [5] F. Xia et al. "Mobile Multimedia Recommendation in Smart Communities", School of Software, Dalian University of Technology, Dalian 116620, China, VOLUME 1, PP.19, 2013.
- [6] Baltrunas L. et al. "InCarMusic: Context-Aware Music Recommendations in a Car", LNBIP 85, pp. 89–100, Springer-Verlag Berlin Heidelberg 2011.
- [7] Internet: "What is Research?" <https://ori.hhs.gov/education/products/sdsu/variables.htm>
- [8] DFID, "What is the evidence on the impact of research on international development", UK Department for International Development: Leading the UK government's fight against world poverty, V.1.1, July 2014.
- [9] Adrienn CZÁRL and Mária BELOVECZ, "Role of Research and Development in the 21st Century", Revista Informatica Economică, nr. 4 (44), 2007.
- [10] Andr e Vellino and David Zeber, "A Hybrid, Multi-Dimensional Recommender for Journal Articles in a Scientific Digital Library", Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology, pp.111-114, 2007.
- [11] Kwanghee Hong, Hocheol Jeon, and Changho Jeon, "Personalized Research Paper Recommendation System using Keyword Extraction Based on UserProfile", Journal of Convergence Information Technology (JCIT), Volume8, Number16, November 2013.
- [12] Kazunari Sugiyama and Min-Yen Kan, "Scholarly Paper Recommendation via User's Recent Research Interests", JCDL'10 Proceedings of the 10th annual joint conference on Digital libraries, 2010.
- [13] Christoffer Davidson, "Mobile Application Recommender System", UPTEC IT 10 025, Examensarbete 30 hp, December 2010.
- [14] Bela Gibb et al. "Scienstein: A Research Paper Recommender System", CONFERENCE PAPER, JANUARY, 2009.
- [15] N. Agrawal et al. "Research Paper Recommender Systems: A Subspace Clustering Approach", LNCS 3739, pp. 475–491, 2005.
- [16] J. Serrano-Guerrero et al. "A google wave-based fuzzy recommender system to disseminate information in University Digital Libraries 2.0", Information Sciences 181, 1503-1516, 2011.
- [17] Don Conry, et al. "Recommender Systems for the Conference Paper Assignment Problem", RecSys'09, October 23–25, 2009, New York, USA.
- [18] Damianos Gavalas and Michael Kenteris, "A web-based pervasive recommendation system for mobile tourist guides", Pers Ubiquit Comput15:759–770, 2011.
- [19] Joeran Beel, et al. "Research Paper Recommender Systems: A Literature Survey" International Journal on Digital Libraries, Springer Berlin Heidelberg, 26 Jul 2015. DOI 10.1007/s00799-015-0156-0.
- [20] H. J. Lee and Sung Joo Park, "MONERS: A news recommender for the mobile web", Expert Systems with Applications 32:43–150, available at "www.sciencedirect.com", 2007.
- [21] Tim Hussein, et al. "Hybreed: A software framework for developing context-aware hybrid recommender systems", User Model User-Adap Inter 24:121–174, DOI 10.1007/s11257-012-9134-z, 2014.
- [22] Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, Collaborative Filtering Recommender Systems. Foundations and Trends R in Human–Computer Interaction Vol. 4, No. 2 (2010) 81–173, 2011, DOI: 10.1561/1100000009.
- [23] Pinata Winoto, Tiffany Ya Tang and Gordon McCalla, Contexts in a Paper Recommendation System with Collaborative Filtering. Konkuk University of Korea.
- [24] D. Gavalas et al. / Journal of Network and Computer Applications 39 (2014) 319–333.
- [25] Hevner et al./Design Science in IS Research. MIS Quarterly Vol. 28 No. 1, pp. 75-105/March 2004.
- [26] J ran Beel, Towards Effective Research-Paper Recommender Systems and User Modeling based on Mind Maps. Otto-von-Guericke-University Magdeburg.
- [27] Donald C. Conry, Recommender Systems for the Conference Paper Assignment Problem. June 4, 2009 Blacksburg, Virginia.
- [28] Cheng Long, Raymond Chi-Wing Wong, and Yu Peng, On Good and Fair Paper-Reviewer Assignment. 2013 IEEE 13th International Conference on Data Mining, Hong Kong University of Science and Technology Kowloon, Hong Kong.
- [29] Xiang Liu, Torsten Suel, and Nasir Memon, A Robust Model for Paper-Reviewer Assignment. San Jose or vicinity, CA, USA Copyright 2014.
- [30] Mooney, R. J. and Roy, L. 2000. Content-based book recommending using learning for text categorization. *Proceedings of the fifth ACM conference on Digital libraries* (2000), 195–204.
- [31] Brusilovsky, P., Farzan, R. and Ahn, J. 2005. Comprehensive personalized information access in an educational digital library. *Digital Libraries, 2005. JCDL'05. Proceedings of the 5th ACM/IEEE-CS Joint Conference on* (2005), 9–18.
- [32] Cesar Vialardi et al. Recommendation in Higher Education Using Data Mining Techniques. Educational Data Mining 2009, university of Autonoma, Madrid.
- [33] Andreas G. Educational and scientific recommender systems: Designing the information channels of the virtual university. 2001, Great Britain.
- [34] Faensen, D., Faultstich, L., Schweppe, H., Hinze, A. and Steidinger, A. 2001. Hermes: a notification service for digital libraries. *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries* (2001), 373–380.
- [35] Bradshaw, S. 2003. Reference directed indexing: Redeeming relevance for subject search in citation indexes. *Research and Advanced Technology for Digital Libraries*. (2003), 499–510.
- [36] Ritchie, A. 2008. *Citation context analysis for information retrieval*. PhD Research work. University of Cambridge.

- [37] Ritchie, A., Teufel, S. and Robertson, S. 2008. Using terms from citations for IR: some first results. *Advances in Information Retrieval*. Springer. 211–221.
- [38] Radhya Sahal1, SaharSelim1and Abeer ElKorany, An Adaptive Framework for Enhancing Recommendation Using Hybrid Techniques. *International Journal of Computer Science & Information Technology (IJCSIT)* Vol 6, No 2, April 2014.
- [39] Adomavicius, G. & Tuzhilin, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions', *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734-749, 2005.
- [40] Fan, H. & Poole, M. S. 'What is personalization? Perspectives on the design and implementation of personalization in information systems', *Journal of Organizational Computing and Electronic Commerce*, vol. 16, no. 3-4, pp. 179-202, 2006.
- [41] Michael Steinbach George Karypis and Vipin Kumar, A Comparison of Document Clustering Techniques. Department of Computer Science and Engineering, University of Minnesota.
- [42] Pham M. C., Cao Y., Klamma R., Jarke M. A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis. *Journal of Universal Computer Science*, vol. 17, no. 4, 2011.
- [43] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*, 2004.
- [44] Malak Al-Hassan, Semantic-Enhanced Hybrid Recommender Systems for Personalised E-Government Services. Engineering and information technology university of technology, Sydney, Australia, 2014.
- [45] Call for Papers: International Conference on Educational Research for Development. Designing and developing a mobile based tourist recommender system: By: The College of Education, Addis Ababa University, Ethiopia 13th-15th, on May 2009.
- [46] American Scientific Research Journal for Engineering, Technology, & Sciences (ASRJETS). Retrieved at <http://publicationethics.org/resources/flowcharts> on January 2016.
- [47] John MacColl, Supporting Research: Environments, Administration and Libraries. European Director, RLG Partnership OCLC Research.
- [48] Betsy Kulamer et al. Recommended Practices to Ensure Technical Conference Content Quality. 4th World Conference on Research Integrity Rio De Janeiro, Brazil, June 2, 2015.
- [49] John Benjamin Schafer, MetaLens: A Framework for Multi-source Recommendations. University of Minnesota 2001.
- [50] Simen Fivelstad Smaaberg, Context-Aware Group Recommendation Systems. Norwegian University of Science and Technology, Department of Computer and Information Science, June 2014.
- [51] Andargachew Asfaw, Designing and Developing a Mobile Based Tourist Recommender System: a Hybrid Approach. University of Gondar, JUNE 2015.
- [52] F. Ricci, L. Rokach, B. Shapira, and K. B. P., *Recommender systems handbook*. Springer, 2011, pp. 1–35.
- [53] J. Beel, S. Langer, M. Genzmehr, and A. Nürnberger, "Introducing Docear's Research Paper Recommender System," in *Proceedings of the 13th ACM/IEEECS Joint Conference on Digital Libraries (JCDL'13)*, 2013, pp. 459–460.
- [54] K. Jack, "Mendeley: Recommendation Systems for Academic Literature," Presentation at Technical University of Graz (TUG). 2012.
- [55] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*, Morgan Kaufmann, 2005.
- [56] Forman, G.: An Extensive Empirical Study of Feature Selection Metrics for Text Classification. *Journal of Machine Learning Research* 3 (2003) 1289–1305.
- [57] Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management* 24 no. 5 (1988) 513–523.
- [58] Christoffer Davidsson, Mobile Application Recommender System. ISSN: 1401-5749, UPTec IT 10 025, December 2010.
- [59] S. L. Ting, W. H. Ip, and Albert H. C. Tsang, Is Naïve Bayes a Good Classifier for Document Classification? *International Journal of Software Engineering and Its Applications*, Vol. 5, No. 3, July, 2011.
- [60] Rich, "User modeling via stereotypes," *Cognitive science*, vol. 3, no. 4, pp. 329–354, 1979.
- [61] Ken Peffers et al. Design Science Research Evaluation. Lee Business School, University of Nevada Las Vegas, Las Vegas NV USA.
- [62] Montaner, M., López, B., Lluís de la Rosa, J. Taxonomy of Recommender Agents on the Internet. In: *Artificial Intelligence Review*, vol. 19, no. 4, 2003. Kluwer Academic Publishers, Netherlands (2003) pp. 285-330.
- [63] Inderjit S. Dhillon, James Fan and Yuqiang Guan, Efficient clustering of very large document collections.
- [64] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. University at Dortmund, Germany.
- [65] Mortensen, Magnus. "Design and evaluation of a recommender system." (2007).
- [66] Burke, Robin. "Hybrid Web Recommender Systems". School of Computer Science, Telecommunications and Information Systems. DePaul University, 243 S. Wabash Ave. Chicago, Illinois, USA.
- [67] John Gossman. Introduction to model/view/viewmodel pattern for building wpf apps. <http://blogs.msdn.com/b/johngossman/archive/2005/10/08/478683.aspx>; retrieved on September 2016.
- [68] Alberto H. F. Laender et al. A source independent framework for research paper recommendation. *JCDL'11*, June 13–17, 2011, Ottawa, Ontario, Canada, 2011.