

Guide Me: A Research Work Area Recommender System

Richa Sharma

CSE Department, Chandigarh University, Gharuan, Mohali, 140413, India

E-mail: richas30@yahoo.com

Sharu Vinayak, Rahul Singh

CSE Department, Chandigarh University, Gharuan, Mohali, 140413, India

E-mail: sharuvinayak@gmail.com, rahulsinghce25@gmail.com

Abstract—With the advent of Industrial Revolution, not only the choices in various fields increased but also the era of computer came into existence thereby revolutionizing the global market. People had numerous choices in front of them that often led to the confusion about what product might actually fulfill their requirements. So the need for having a system which could facilitate the selection criteria and eradicate the dilemma of masses, was realized and ultimately recommender systems of present day world were introduced. So we can refer recommender systems as software tools that narrow down our choices and provide us with the most suitable suggestions as per our requirements. In this paper, we propose a novel recommender system i.e. RWARS (Research Work Area Recommender System) that will recommend research work area to a user based on his/her characteristics similar to those of other users. The characteristics considered here are hobbies, subjects of interests, programming skills and future objectives. The proposed system will use Cosine Similarity approach of Collaborative Filtering.

Index Terms—Collaborative Filtering, Cosine Similarity, Recommender Systems.

I. INTRODUCTION

The term Recommender System is a combination of two words: Recommender and System, where the previous one derived from the Latin word “Recommendere” means- to praise or commend to other as being worthy or desirable, to endorse or to entrust; while the later derived from the Latin word “Systema” means a group of interacting, inter-related or independent elements forming a complex whole. So we can say that recommender systems help making the selection process easier and time saving. The aim of Recommender Systems is to provide the user with the most suitable recommendation, from the plethora of options available.

Recommender systems can be referred to as a system having users, a user interface, a dataset and various recommendation algorithms. It would not be wrong to say that the “Recommender systems are- for the user, to the

user and by the user.” It can be elaborated from the fact that the recommendations are made for the users and the dataset used for making the recommendations is also taken from the users themselves in the form of ratings, which can both be implicit or explicit. Then we have the user-interface which acts as the intermediate between the user and the system. The user expresses his preferences, ratings or requirements through the user-interface only and the recommendations are also displayed on the same. Finally, once the user requirements are gathered, some recommendation algorithm is applied. Till date, we have plenty of recommendation algorithms ranging from approach to approach (Table 2). Fig.1 gives an overview of how various aspects of recommender system are related to each other.

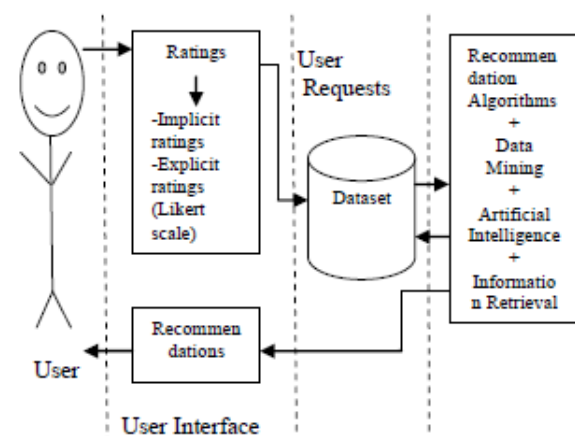


Fig.1. Recommender Framework

Recommender systems might be a new concept in research area, but it has been prevailing in the society since ages. The concept of recommendations could be seen in case of cavemen, ants and other creatures too. People now have numerous choices in front of them that often lead to the confusion about what product might actually fulfill their requirements. And to overcome this dilemma, we have Recommender systems. Recommender systems can be referred to as software tools and techniques which narrow down our choices of products or items and suggest the most suitable ones based on the

implicit or explicit information. The basic idea that lead to the development of Recommender System is that: we often rely on recommendations from our peers before trying out something new e.g.: while buying mobile phones, checking reviews on book my show app before going for movies, even before visiting a doctor and so on. Although there were no recommender systems in old times, still recommendations were there. Kings used to get their recommendations from the ministers e.g. Birbal in case of King Akbar, acted as a recommender system, giving him suggestions regarding the kingdom’s policies and important decisions. The most well-known recommender systems of old times were the family relatives suggesting match for marriages. So the concept of recommender systems is not new in real-world scenario, rather it just got automated with advancement in technology and got much better and efficient saving our time.

Researchers started working on Recommender systems as an independent research area during the mid 1970s in Duke University. Since then this area has proven to be the centre of attraction for many researchers and a lot of development has been made in numerous domains using various recommendation approaches developed in the meantime. The first commercial Recommender System was Tapestry, developed in 1992 by Xerox Palo Alto Research Centre [5]. The motivation behind it was the increasing use of e-mail, as a result of which users were being inundated by a huge stream of incoming documents. It used Collaborative filtering.

There are various Recommendation approaches developed so far: Collaborative Filtering, Content-based Filtering, Knowledge-based systems, Hybrid systems, Demographic recommender systems and Community-based systems. Table.1 gives an overview of the most commonly used recommendation approaches.

Table 1. Recommendation Approaches

S.no.	Approach	Pros and cons
1.	<p>Collaborative Filtering</p> <ul style="list-style-type: none"> i. This is basically people-to-people co relation. ii. It uses the wisdom of the crowd to recommend items to the user. iii. The ratings can either be explicit or implicit. iv. It assumes that people who had similar tastes in past, will also have similar tastes in future also. 	<p>Pros:</p> <ul style="list-style-type: none"> i. Provides better results when the no. of users and the no. of ratings are available in abundance. ii. No knowledge engineering required. <p>Cons:</p> <ul style="list-style-type: none"> i. Cold-start problem both in case of new users and new items. ii. Sparsity problems
2.	<p>Content based Filtering</p> <ul style="list-style-type: none"> i. It is based on the concept “Show me more of what I have liked”. ii. It takes into account user preferences and on the basis of that, makes the recommendations. iii. It recommends those items which are similar to the user preferences based on their past behavior. 	<p>Pros:</p> <ul style="list-style-type: none"> i. It does not rely on other users’ preferences. ii. Comparison between items possible <p>Cons:</p> <ul style="list-style-type: none"> i. Cold start for new users. ii. No surprises. iii. Grey-sheep problem
3.	<p>Knowledge based</p> <ul style="list-style-type: none"> i. It is based on the concept “Tell me what fits my needs”. ii. No user profile is maintained, rather user interaction is there. iii. It uses knowledge of users & items. 	<p>Pros:</p> <ul style="list-style-type: none"> i. Deterministic recommendations ii. Assured quality iii. No cold start <p>Cons:</p> <ul style="list-style-type: none"> i. Cost of knowledge acquisition from domain experts, users etc. are too high. ii. Basically static
4.	<p>Hybrid Recommender Systems</p> <ul style="list-style-type: none"> i. A hybrid system combining techniques A and B tries to use the advantages of A to fix the disadvantages of B. 	<p>Pros:</p> <ul style="list-style-type: none"> i. Avoids most of the shortcomings of other approaches. <p>Cons:</p> <ul style="list-style-type: none"> i. A little complex to apply on the same dataset.

Table 2. Recommendation Algorithms

S.No.	Algorithms
1.	<p>Pearson co-relation is used to find similarity between users based on their ratings.</p> $sim(a,b) = \frac{\sum_{p \in P} (r_{a,p} - \bar{r}_a)(r_{b,p} - \bar{r}_b)}{\sqrt{\sum_{p \in P} (r_{a,p} - \bar{r}_a)^2} \sqrt{\sum_{p \in P} (r_{b,p} - \bar{r}_b)^2}}$ <p>Here, <i>a, b</i> - users <i>r_{a,p}</i> - rating of user <i>a</i> for product <i>p</i> \bar{r}_a, \bar{r}_b - average user rating <i>P</i> - set of products rated by both <i>a</i> and <i>b</i></p>
2.	<p>Constrained Pearson co-relation is also used to find similarity between users, when co-relation between them increases.</p> $sim(a,b) = \frac{\sum (p_{ai} - mp)(p_{bi} - mp)}{\sqrt{\sum (p_{ai} - mp)^2} \sqrt{\sum (p_{bi} - mp)^2}}$ <p>Here, <i>mp</i> - midpoint of rating scale</p>
3.	<p>Cosine similarity is a vector based approach based on linear algebra.</p> $sim(a,b) = \frac{r_a \cdot r_b}{\ r_a\ _2 \ r_b\ _2}$
4.	<p>Spearman Co-relation is a non-parametric method that computes a measure of co-relation between ranks.</p> $sim(a,b) = \frac{\sum (rank_{ai} - \overline{rank_a})(rank_{bi} - \overline{rank_b})}{\sqrt{\sum (rank_{ai} - \overline{rank_a})^2} \sqrt{\sum (rank_{bi} - \overline{rank_b})^2}}$
5.	<p>K Nearest neighbour approach is a simple algorithm used as a non-parametric technique to calculate the similarity measure or distance between the neighbors (users, items or products). Nearest neighbor can be found out by calculating the distance between all the neighbors by using:</p> <ul style="list-style-type: none"> • Euclidean distance $\sqrt{\sum_{i=1}^k (x_i - y_i)^2}$ <ul style="list-style-type: none"> • Manhattan distance $\sum_{i=1}^k x_i - y_i $

From social networking sites to marriage portals, from e-commerce sites to movie, music or book recommender system, we have recommendations for almost every field. They have a really crucial role in internet sites like Amazon, Pandora, LinkedIn, Netflix, Matrimonial sites, CiteSeer, Social networking sites, YouTube, Yahoo, Jester and a lot more. These websites use either one of the recommendation approach or a combination of two or more. For example, Amazon uses Item to Item collaborative filtering and content-based filtering, CiteSeer uses collaborative and content-based filtering, Pandora uses content-based filtering, Youtube uses personalized recommendations while both Jester and LinkedIn use collaborative filtering. Recommender Systems have evolved so much over the past two decades and now have included Artificial intelligence, Information retrieval and Human-computer interaction in their journey [1], and hence have got more efficient and have gained more popularity.

Table 3. Some common Recommender Systems developed so far

S.No.	Recommender system	Approach
1.	Facebook	Collaborative filtering Content based
2.	Twitter	Collaborative filtering
3.	LinkedIn	Collaborative filtering
4.	Netflix	Hybrid recommender system
5.	MovieLens	Collaborative filtering
6.	Pandora	Content-based recommendation
7.	Last.fm	Collaborative filtering
8.	Amazon	Item to item collaborative filtering, Content-based recommendation
9.	eBay	Collaborative filtering Demographic recommendation
10.	CiteSeer	Content-based Collaborative filtering
11.	Docear	Content-based
12.	Techlens	Content-based Collaborative filtering
13.	Youtube	Personalized recommendation
14.	Jester	Collaborative filtering

Since, Recommender Systems have reduced our effort of going through each and every item and selecting the most suitable one, it would be great if there is a Recommender System that can suggest an area to a student that he or she can do his research in. In this paper, we propose RWARS (Research Work Area Recommender System) that recommends an area for research work by using cosine-similarity. This system would find the similarity between the users in terms of their future objectives, the type of research they want to do, their hobbies, subjects of interests and their programming skills. As the dataset, we have collected our own data by conducting a small survey of Post-graduate students who have already worked in some research area. We anticipate that our proposed system will prove to be beneficial.

This paper is organized as follows. Section II evaluates the various works relative to the proposed system. Section III reports the framework of the proposed system. In section IV, various aspects of the system are discussed and the processes in each phase are elaborated. Finally, the paper is concluded in Section V.

II. RELATED WORK

In this section, we briefly describe some of the work related to recommender systems in general, recommender systems related to the field of education and recommender systems using collaborative filtering, keyword extraction and works that have deployed cosine similarity.

Recommender system is one of the most popular and mighty tool, that has taken over e-commerce [1]. The basic idea that led to its development is to help the users select the most suitable product or item from a plethora of options in front of them [1] [2]. The most popular approaches of Recommender Systems are: Collaborative Filtering, Content-based systems, Hybrid systems and Knowledge-based systems.

Collaborative filtering is also known as ‘people-to-people co-relation’ [1]. It is based on a simple idea that users having identical interests in the past, tend to have alike interests in the future also [1] [2] [3] [4]. Content-based systems follow the concept “Show me more of what I have liked” [4]. The basic idea is to recommend products to a user, which are identical to the ones that user has already liked in the past [1] [2]. The similarity between two or more items can be calculated using the features they have in common.

Knowledge-based systems follow the concept “Tell me what fits my needs” [4]. Such systems use domain knowledge of a specific product to make recommendations to the user [1]. The system gets the user requirements, matches it with its knowledge base and makes the recommendations of the most appropriate and suitable product to the user, keeping user preferences in consideration. Hybrid system gives the best of both worlds. As the name suggests, these systems combine the features of two recommendation approaches. The basic purpose of these systems is to combine the features of two systems (recommendation approaches) in such a way

that the shortcomings of one are overcome by the other [1] [2] [4].

Collaborative filtering suffers from cold-start problem while there is no element of surprise in case of content-based systems [4]. So combining the features of these two approaches often prove to be more efficient and useful. Using the features of explicit ratings and implicit ratings of collaborative filtering and content based systems respectively, the problem of data-sparsity and cold-start problem can be tackled. This approach was used by Mojtaba *et al.* [6] in their work where, they presented a recommender system for recommending learning material to the users. Their work deployed genetic algorithm along with a multi-dimensional information model. Joonseok *et al.* [7] proposed a Personalized Academic Research Paper Recommender system that used collaborative filtering. The system used web-crawler to retrieve various research papers from the internet, and then found the similarity between two research papers using Nearest neighbor approach and finally made the recommendations. Similar work was forth put by Nitin *et al.* [8] who too worked for reducing the sparsity of the data. For that, they introduced Subspace Clustering Algorithm (SCuBA) for recommending research papers to the user.

Nazpar Yazdanfar *et al.* [9] proposed a Link Recommender system that recommends URLs to twitter users using nearest neighborhood approach. They considered hashtags as the representatives of the URLs and used Euclidean, Jaccard, Dice coefficient and Cosine similarity measures to find the item-hashtag and item-user similarity. Pu Wang [10] presented a personalized collaborative approach that was based on the clustering of the customers to overcome the scalability issue. Approaches like K-means clustering, Cosine and Correlation similarity and Nearest neighbor approach were deployed and along with scalability, issues like data sparsity and cold-start problem were also resolved.

Kwanghee *et al.* [11] proposed a personalized research paper recommender system that recommends research paper to the user based on his profile. The system used the concept of keyword extraction to find the similarity between the topics that the user has desire to work in and the research papers available on the web. To find the similarity, Cosine similarity was used and this system proved to be quite efficient in making the search for research papers easier.

One of the newest works being done in Recommender systems is that of cross-domain recommendation i.e. collaborating two different domains together and making the recommendations based on the relation between the two. Hla Hla Moe *et al.* [12] proposed an approach for building ontologies using case-based approach for cross-domain recommendation for cosmetics related to various skin problems. For this work, Ford-Fulkerson algorithm has been used and the results showed that the system proved to be quite efficient and gave better recommendations as compared to other cross-domain systems. Naziha Abderrahim *et al.* [13] proposed a social-trust aware system that based on the social qualities of web services, recommended web services to

the user taking their overall experience into consideration.

Fatiha Bousbahi *et al.* [14] proposed a recommender system for online courses to the user based on his requests. The proposed system used case-based approach for making the recommendations to the user as per his user profile, knowledge and requirements. Haifeng Liu *et al.* [15] presented a novel model for calculating the new user similarity. The work focused primarily on improving the quality of recommendations in case of insufficient user-ratings. Diksha Nagpal *et al.* [16] proposed a faculty recommender system. This system used both user-user collaborative filtering and item-item collaborative filtering. Results of the study revealed that item-item collaborative filtering performed better than user-user collaborative filtering.

Based on the previous work done, we concluded that some more work can still be done in the field of education by developing a Research Work Area Recommender System. We have article recommender system, recommender system for digital libraries and even research paper recommender systems. But selecting what research area to work on is in itself a very tedious task as it requires going through a lot of research works to decide which one interests you the most. With the proposed system, we can not only save time and effort, but we can also make this selection process a lot easier and simpler. The user has to simply answer a short questionnaire and in no time, he or she would be provided with the most suitable recommendations. So we propose RWARS (Research Work Area Recommender System) that would recommend research work areas to the users based on their interests, hobbies, programming skills, subjects of interests and future objectives.

III. SYSTEM ARCHITECTURE

RWARS (Research Work Area Recommender System) is composed of a User Interface, Extractor, Filter, Profile Manager and DB Manager.

Figure 1 gives an overview of the architecture of the proposed system.

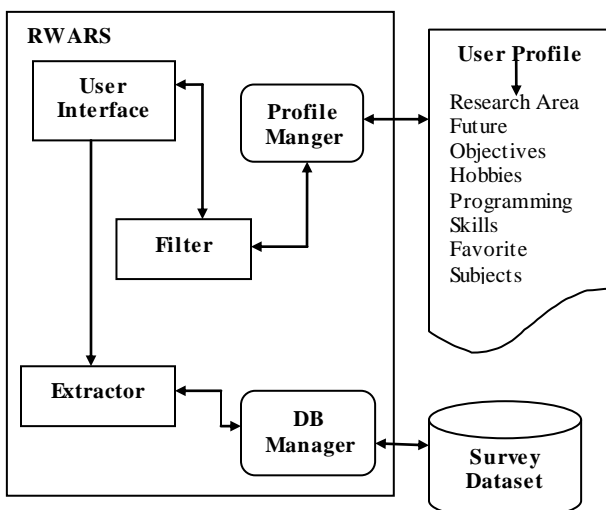


Fig.2. System Architecture of RWARS

The User Interface interacts with the users, takes the input information and sends it to the Extractor that extracts the words or the terms. The basic purpose is to first extract the words from the form filled by the new user and then extract similar words from the dataset. We have designed a very simple user-interface using ASP.NET. The User Interface is composed of a simple form asking the user questions similar to the ones that were asked in the survey. All the user has to do is answer the questions asked in the form and then click on the submit button. Once that is done, the extractor would come into action and extract all the words i.e. whatever his/her responses to the questions are. So the hobbies, subjects of interests, future objectives and programming skills as selected by the user, would be the words or terms to be extracted.

Filter undergoes filtering process to make sure that only the relevant data are taken. Relevant data in our case is only the data that are to be compared. For example, we are only considering the hobbies, subjects of interests, skills and the future objectives. Parameters like name or contact information are not being taken into account.

User Profile stores the information of the users along with the recommendations made. For each user, who would use the system, the results would be saved along with the other data (as filled by him/her in the form). This way a user can keep track of recommendations made to him, based how he filled the form. Also the user can see how his varying interests in subjects or how his programming skills affect the recommendations made. It does not at all depend on an individual that what recommendations would be made, rather his characteristics and skills would be the deciding factor. For example, consider a user "x". Now the first time he uses the system, he is recommended Data Mining, suppose. But the next time he is recommended Recommender Systems. Now it does not imply that the system does not work effectively, rather it means that the second time when the user filled the data, he might have learned some new programming skill or he might have developed his interest in a new subject or he might have changed his objective. So for each user or a user with varying interests, the results would vary.

The DB Manager manages the dataset that we have collected after conducting the survey of students of Masters Degree. The survey had simple questions asking the students about their Research area, Future objectives (job or PhD), Hobbies, Programming skills, what kind of research they wanted to do (Academic or Industrial) and if they would simply opt for the research area suggested by the teacher they wish to work with.

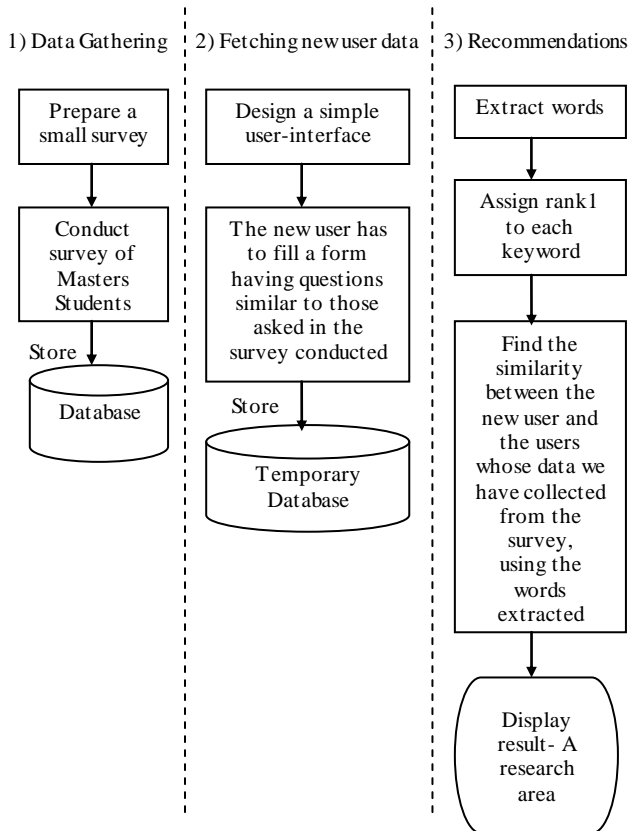


Fig.3. Process of the proposed system

IV. RESEARCH WORK AREA RECOMMENDATION

The proposed recommender system will use Case-based approach to find the similarity between a user who wishes to get the recommendations and the ones who have already opted for some research work area. As shown in Fig.3, RWARS is divided into three phases: Data gathering, Fetching new user data and Making recommendations.

A. Data gathering

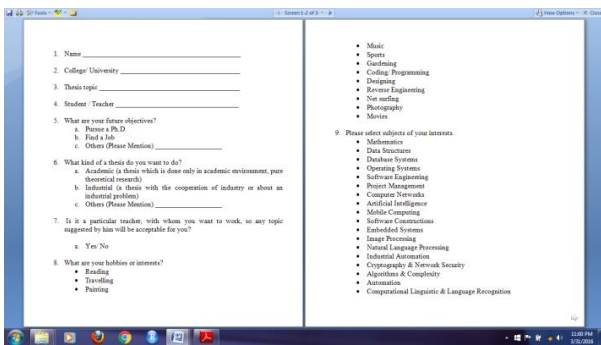


Fig.4. Survey Form

For the database of the proposed system, we have first prepared a small survey and then asked the Post-graduate students to fill the survey. The data gathered from the survey is entered into the system manually and will be

used as the dataset. Fig.4 and Fig.5 shows the questions that were asked in the survey while in Fig.6, we have a .CSV file of the data that have been gathered.

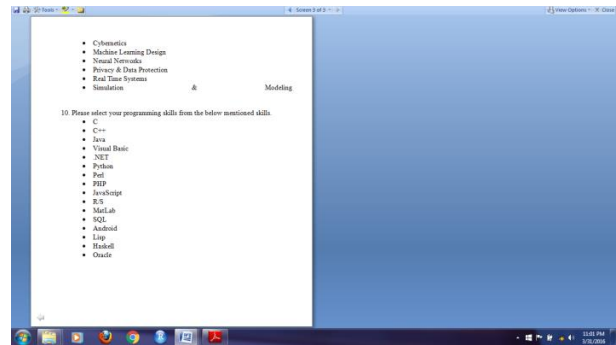


Fig.5. Survey Form

AREA OF THIS	TEACHER SUGGESTED/NOT	THESE KIND	OBJECTIVES	HOBBIES	FAVORITE SUBJECTS
1. Area of this	Academic	Job	Reading, Music, Movies	Reading, Music, Movies	Data structures, Operating systems, Soft
2. Recommender Systems	Academic	Job	Music, Travelling, Reading, Photography	Music, Travelling, Reading, Photography	Software Engineering, Project Manage
3. Digital Image Processing	No	Academic	Reading, Music, Painting, Net surfing, Photography	Reading, Music, Painting, Net surfing, Photography	Operating systems, Software engineeri
4. Recommender Systems	No	Academic	Reading, Travelling, Music, Gardening, Reverse Engineering, Net surfing, Photography	Reading, Travelling, Music, Gardening, Reverse Engineering, Net surfing, Photography	Operating systems, Software engineeri
5. Software Bitmapping	No	Academic	Job	Music, Photography, Movies	Operating systems, Software engineeri
6. Recommender Systems	No	Academic	Job	Travelling, Music, Designing, Reverse engineering, Photography	Mathematics, Data structures, Databas
7. Reverse Engineering	Yes	Academic	PHD	Reading, Painting, Music, Coding, Designing, Net surfing, Photography, Movies	Mathematics, Data structures, Comput
8. Networking	Yes	Academic	PHD	Painting, Music, Designing, Net surfing, Photography, Movies	Data structures, Cybernetics, Mathwa
9. Web mining	Yes	Academic	PHD	Reading, Music, Movies, Net surfing, Photography	Software Engineering, Project Manage
10. Data Forensics	Yes	Academic	Job	Reading, Travelling	Mathematics, Database systems, Soft
11. Web analysis & Reusability	No	Academic	Job	Music, Coding, Reading	Database systems, Software engineeri
12. Data mining	No	Industrial	Job	Music, Photography, Movies, Net surfing	Mathematics, Database systems, Soft
13. Web management	No	Academic	Job	Sports, Net surfing	Mathematics, Database systems, Soft
14. Recommender Systems	No	Academic	Job	Reading, Net surfing	Mathematics, Database systems, Soft
15. Natural Language Processing	No	Academic	Job	Music, Movies	Data structures, Database systems, Soft
16. Natural Language Processing	No	Industrial	Job	Music, Movies, Net surfing	Operating systems, Software engineeri
17. Data mining	No	Academic	Job	Travelling, Music, Movies	Operating systems, Software engineeri
18. Digital Image Processing	Yes	Academic	Job	Travelling, Music, Gardening	Software Engineering, Project Manage
19. Cloud computing	Yes	Academic	Job	Travelling, Music, Gardening, Movies	Software Engineering, Project Manage
20. Cloud computing	Yes	Academic	Job	Travelling, Painting, Designing, Movies	Software Engineering, Project Manage
21. Cloud computing	Yes	Academic	Job	Travelling, Music, Gardening, Photography	Software Engineering, Project Manage
22. Digital Image Processing	No	Academic	Job	Travelling, Music	Mathematics, Software engineering, In
23. Digital Image Processing	No	Academic	Job	Travelling, Music	Software engineering, Artificial intell
24. Web analysis	No	Academic	PHD	Travelling, Music	Mathematics, Database systems, Soft
25. Android security	No	Academic	Job	Reading, Travelling, Music, Reverse engineering, Movies	Mathematics, Database systems, Soft
26. Data mining	Yes	Industrial	Job	Music, Designing	Database systems, Software engineeri
27. Cloud computing	Yes	Industrial	Job	Reading, Music, Designing	Software engineering, Computer netw
28. Cloud computing	Yes	Industrial	Job	Reading, Music, Designing	Software engineering, Mobile comput
29. Digital Image Processing	Yes	Academic	Job	Reading, Music, Designing, Net surfing	Mathematics, Data structure, Project M
30. Wireless sensor networks	No	Industrial	Job	Music, Net surfing	Computer networks, Mobile computi
31. Digital Image Processing	No	Industrial	Job	Reading, Painting, Music, Coding, Designing, Net surfing, Photography	Mathematics, Data structure, Project M
32. NET	Yes	Academic	PHD	Travelling, Photography	Software engineering, Computer netw
33. NET	Yes	Industrial	PHD	Reading, Music, Sports, Coding, Designing, Reverse engineering	Software engineering, Computer netw
34. NET	No	Academic	Job	Painting, Music, Coding, Designing, Reverse engineering	Software engineering, Computer netw

Fig.6. Dataset gathered from the survey

B. Fetching new user data

We have designed a simple User-interface, where a new user would fill the form as shown in Fig.7. Once the user fills the form and clicks on the submit button, his information would be stored in a temporary database. All the algorithms would work at the back-end and once this is done, the results would be displayed on the user-interface only.

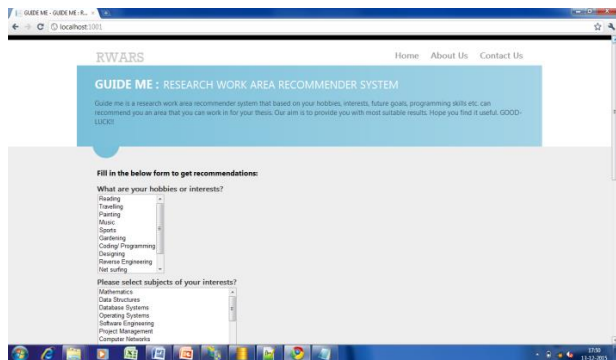


Fig.7. User-Interface

C. Making Recommendations

As shown in Fig.2, for making the recommendations first the Extractor would extract all the words from the Temporary dataset. For convenience, we would assign rank 1 to each word. Once this is done, we would find out the similarity between the new user and the ones in the

dataset using Cosine similarity. It is a vector based approach based on linear algebra. Users are considered as vectors and similarity between them is measured by the cosine distance or dot product of the angles between the rating vectors [2] [3]. It is given by:

$$\text{sim}(a, b) = \frac{r_a \cdot r_b}{\|r_a\|_2 \|r_b\|_2} \quad (1)$$

In Equation 1,

r_a is the rating given by user a

r_b is the rating given by user

In the proposed work, we would consider the new user as user a and rest all the users of the dataset would be considered as user b. Then we would apply the cosine similarity algorithm to find the similarity between the users (a and b). Finally, for a user a having maximum similarity with a user b, recommendation for the research work area would be made.

V. CONCLUSION

In this paper, we proposed a novel recommender system RWARS that will recommend a research work area to the user based on the similarity of user characteristics between the new user and the ones who have already worked in some research area. A user has to answer a simple questionnaire asking him about his skills, subjects of interests, future objectives etc. and based upon these answers he will be provided with the recommendations.

The proposed system is still in its development stage and a lot of work still needs to be done. We have conducted a survey of Masters Students and used that as the dataset. Also, at present we have planned to use Cosine-similarity approach only to make the recommendations. But in future we aim to apply other recommendation approaches too and then compare the results obtained from each approach. The parameters that would be considered while deciding which approach is better than the other are coverage, precision, recall and effectiveness. Now just the name of the research area would not be sufficient for the user to take any decision, so in future, we have planned to give the links of a literature review of the research areas too along with the names thereby helping the user to the maximum.

REFERENCES

- [1] Ricci, Francesco, Lior Rokach, and Bracha Shapira. *Introduction to recommender systems handbook*. Springer US, 2011.
- [2] Jannach, Dietmar, Markus Zanker, Alexander Felfelnic, and Gerhard Friedrich. *Recommender systems: an introduction*. Cambridge University Press, 2010.
- [3] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan. "Collaborative filtering recommender systems." *Foundations and Trends in Human-Computer Interaction* 4, no. 2 (2011): 81-173.
- [4] Jannach, Dietmar, and Gerhard Friedrich. "Tutorial:

recommender systems." In *Proceedings of the International Joint Conference on Artificial Intelligence, Barcelona*. 2011.

- [5] Huttner, Joseph. "From Tapestry to SVD: A Survey of the Algorithms That Power Recommender Systems." (2009).
- [6] Salehi, Mojtaba, Mohammad Pourzaferani, and Seyed Amir Razavi. "Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model." *Egyptian Informatics Journal* 14, no. 1 (2013): 67-78.
- [7] Lee, Joonseok, Kisung Lee, and Jennifer G. Kim. "Personalized academic research paper recommendation system." *arXiv preprint arXiv:1304.5457*(2013).
- [8] Agarwal, Nitin, Ehtesham Haque, Huan Liu, and Lance Parsons. "Research paper recommender systems: A subspace clustering approach." In *Advances in Web-Age Information Management*, pp. 475-491. Springer Berlin Heidelberg, 2005.
- [9] Yazdanfar, Nazpar, and Alex Thomo. "Link recommender: Collaborative-Filtering for recommending URIS to Twitter users." *Procedia Computer Science* 19 (2013): 412-419.
- [10] Wang, Pu. "A Personalized Collaborative Recommendation Approach Based on Clustering of Customers." *Physics Procedia* 24 (2012): 812-816.
- [11] Hong, Kwanghee, Hocheol Jeon, and Changho Jeon. "Personalized Research Paper Recommendation System using Keyword Extraction Based on UserProfile." *Journal of Convergence Information Technology* 8, no. 16 (2013): 106.
- [12] Moe, Hla Hla, and Win Thanda Aung. "Building Ontologies for Cross-domain Recommendation on Facial Skin Problem and Related Cosmetics." *International Journal of Information Technology and Computer Science (IJITCS)* 6, no. 6 (2014): 33.
- [13] Abderrahim, Naziha, and Sidi Mohamed Benslimane. "Towards Improving Recommender System: A Social Trust-Aware Approach." *International Journal of Modern Education and Computer Science (IJMECS)* 7, no. 2 (2015): 8.
- [14] Bousbahi, Fatiha, and Henda Chorfi. "MOOC-Rec: A Case Based Recommender System for MOOCs." *Procedia-Social and Behavioral Sciences* 195 (2015): 1813-1822.
- [15] Liu, Haifeng, Zheng Hu, Ahmad Mian, Hui Tian, and Xuzhen Zhu. "A new user similarity model to improve the accuracy of collaborative filtering." *Knowledge-Based Systems* 56 (2014): 156-166.
- [16] Nagpal, Diksha, Sumit Kaur, Shruti Gujral, and Amritpal Singh. "FR: A Recommender for Finding Faculty Based on CF Technique." *Procedia Computer Science* 70 (2015): 499-507.

Authors' Profiles



Richa Sharma: Belongs to Bilaspur, Himachal Pradesh, India and born on March 6, 1992. She is a Research student for Masters Degree for Computer Science Engineering in Chandigarh University. She received her Bachelor's Degree from Punjab Technical University in 2014. The field of her thesis is Recommender Systems. Her areas of interests for research are R language for

building Recommender Systems, Data Privacy and Security.



Sharu Vinayak: Belongs to Ludhiana, Punjab, India and born on October 8, 1992. She is currently pursuing Masters in Computer Science and Engineering from Chandigarh University and has done Bachelor's degree in Information Technology from Ludhiana College of Engineering and Technology.

Programming and learning about new technologies interests the author.



Rahul Singh: Belongs to Kanpur, Uttar Pradesh, India and born on April 15, 1989. He is an Assistant Professor in CSE Department of Chandigarh University. He has received his Masters degree from Thapar University. His areas of interests are Theory of Computation and Software Engineering.

How to cite this paper: Richa Sharma, Sharu Vinayak, Rahul Singh, "Guide Me: A Research Work Area Recommender System", International Journal of Intelligent Systems and Applications (IJISA), Vol.8, No.9, pp.30-37, 2016. DOI: 10.5815/ijisa.2016.09.04