International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-1, Issue-2, May 2011

An Open Source Framework for PRTF Protocol

Ashok Bhansali, H.R. Sharma

Abstract - **Social networking websites contain vast amount of data inside them. Volume of data is enormous and growing at a very fast rate. Social network data can be classified in three major categories – user profile data, user communication data and group communication data. Data mining can be applied effectively to discover the knowledge and to extract the useful patterns from this gigantic data set, which is called as the social network mining. Person Related to a Field (PRTF) is a protocol to mine the information across all the social networking data, in general, and use the extracted pattern to search an expert in particular. It also proposes the mechanism to rank the searched experts. In this paper we propose an open source implementation of the PRTF (Person related to a Field) protocol. Using this proposed framework, apart from expert identification, number of useful patterns can be discovered from social networking data. The proposed framework is implemented using open source technologies and is explained with the help of an illustrative example.**

Index Terms - **Social network mining, data mining, expert finding, PRTF.**

I. INTRODUCTION

Social networks are the contemporary ways to connect the people across the globe. Social networking websites contain huge amount of data inside them. Volume of data is tremendously huge and growing at a very fast rate. Data mining can be applied effectively to discover the knowledge and to extract the useful patterns from this gigantic data set, which is called as the social network mining.

The social networking data can broadly be classified in three major categories:

A. *User Profile Data* - It includes all the data entered by user for the information of the user. Typically it is the personal information of the user like first name, last name, address, gender, qualification, chat-ids, email-ids etc.

B. *User Communication Data* - The communication data includes many different things. A person communicates with other members on the network on different topics frequently. The important data here is list of the entire directly known and connected person's information and communication data with these connected people. On certain social networking websites the communication can also be done with the non-connected persons.

C. *Group Communication Data* - This is the data which belong to a particular group in the social networking site. All the members of the group can do the communication on some of the topics. This is called as the group communication data.

Manuscript received May 12, 2011

Ashok Bhansali, Research Scholar, Singhania University, Pacheri Bari, Rajasthan, India, Phone: +919827478045, (email: ashok.bhansali@opjit.edu.in).

Dr. H.R. Sharma, Director, Chhatrapati Shivaji Institute of Technology, Durg, C.G., India , (email: hrsharma44@gmail.com).

II. PRTF PROTOCOL

PRTF (Person Relation to a Field) is a protocol used to mine the people's information available across the social networking databases. A person participating in social networking can be a part of many different segments of the social networks, e.g. user can maintain his profile, can participate in a number of associated communication forums/threads/blogs etc. and can be a member of specific groups. The proposed framework integrates different parts/data of the social networks to expand the search operation across all the segments of the social networking databases i.e. user profile, user communication and group databases. It is a search protocol to generate the result as a set of people involved in a particular area or field available across social networking databases and then rank the result according to their expertise level. PRTF algorithm works as below [1]:

```
Algorithm Algorithm for finding a person
related to a field (PRTF)
Require: Profile Data (PDB), Group Data
(GDB), Communication Data (CDB), Search
filed f
1. Extract features from Profile
Database
for all users u in PDB do
      - fetch the list of profile
      entities
      - Get the values of related
      entities ∈ f
      for all entities pe in the profile
      PDB do
      u.score(PDB) += u.score(pe)
      end for
end for

2. Extract features from Group Database
for all users u in GDB do
      - fetch the list of profile
      entities
      - Get the values of related
      entities ∈ f
      for all entities ge in the profile
      PDB do
      u.score(GDB) += u.score(ge)
      end for
end for
3. Extract features from Communication
Database
for all users u in CDB do
      - fetch the list of profile
      entities
      - Get the values of related
      entities ∈ f
```

```
      for all entities ce in the profile
      PDB do
      u.score(CDB) += u.score(ce)
      end for
end for

4. Combine the score of a user from all
the databases.
for all users u do
    u.score = u.score(PDB)+ u.score(PGB)
                          + u.score(PCB)
end for
5. Generate Ranks of the user using the
scores of the users
return list of top ranked users
```

III. PREVIOUS WORK

PRTF (Person Related To a Field) protocol was proposed by Bhansali and Sharma to mine the social network databases to find the expert [1]. In the proposed protocol three major tasks have been focused. First, searching across all the segments of social networking database. Second integrate the various scores and rank the searched result and lastly output representation of the ordered result.

Various issues affects web mining techniques were studied by Ting [2] for analysis of on-line social networks. Techniques and concepts of web mining and social networks analysis were introduced and reviewed along with a discussion about how to use web mining techniques for on-line social networks analysis. Social networks have the surprising property of being "searchable". A model has been presented by Watts, Dodds and Newman [3] that offers an explanation of social network searchability in terms of recognizable personal identities i.e. sets of characteristics measured along a number of social dimensions. An algorithm named ComTector(Com-munity DeTector) was proposed by Du, Wu, Pei, Wang and Xu [4]. The algorithm was proposed to improve the efficiency for the community detection in large-scale social networks based on the nature of overlapping communities in the real world. A number of applications have also been proposed to which ComTector can be applied. The expert finding problem was investigated in detail by Yimam [5] and the existing systems were reviewed and analyzed in this domain and suggested a domain model that can serve as a basis for design and development decisions. An approach was proposed which is a method to generate, maintain and utilize an expertise information space based on dynamic organizational information resources. A methods for finding experts (and their contact details) using e-mail messages was proposed by Balog and Rijke [6]. The messages on a topic are located and then the associated experts are discovered. The task of automatically determining an expert profile of a person from a heterogeneous corpus made up of a large organization's intranet was proposed by Balog and Rijke [7]. A conceptual framework is presented by Breslin, Bojar, Meza, Boley and Mochol [8] for the reuse and interlinking of existing, well-established vocabularies in the Semantic Web. The proposed framework can be used to connect people based on joint or complementing interests. Experts can be discovered using the profiles of people in social networks and using the content they post in online communities. The FindXpRT project for finding experts via rules and taxonomies is developed and proposed by Li, Boley, Bhavsar and Mei [9]. They have implemented rules for a client finding an expert to collaborate with, for an expert's decision making on whether to collaborate and for specifying the collaboration mode. Collaborative environments are only effective when experts are accessible within them and those experts are able and willing to share their knowledge. Jian Jiao and Jun Yan investigated the expertise that users display in online communities, especially in discussion groups and propose an effective expert ranking algorithm, which integrates both discussion thread contents and social network extracted from massive social interactions [10]. They presented a vector space model to compute the content relevance part and a PageRank style algorithm for the expert network part. The algorithm ensures that the highly ranked experts are both highly relevant to the specific queries and highly authoritative in corresponding areas.

IV. PRTF ALGORITHM

The PRTF working mechanism and algorithm is given in by Bhansali and Sharma[1]. This is a five step algorithm. In the first step the searching is done across the user profiles for the field keyword/s. Then in the second and third step searching is done in the Group and Communication (Forum, Discussion etc.) databases of the social networking sites. In the fourth step search result for the first three steps is combined using the common entities available for the search, and finally in the fifth step the result rank is generated on the basis of the combined score of the users.

V. IMPLEMENTATION

The implementation of PRTF algorithm is done using open source technologies Java and MySql. Various table structures for storing social networking data is created in MySql. The following five tables are created to hold various social data as below:

1. Communicationdb - thread_id, from_user, to_user, discussion_word
2. Groupdb – group_id, group_name, description_keywords, group_score
3. Profiledb – user_id, area, experience
4. User_group_mapping – user_id, group_ids
5. Vocabulary – word, word_id

Data Structures Used - Two powerful data structure from java are used for the implementation. The first is Map and the second is Set. Map is used here to store the key value pair for all the users and their corresponding scores for different databases. Set is a collection that contains no duplicate elements and at most one null element. Set is used here to store comma separated values of different fields like group ids, discussion words, description keywords etc.

The stepwise implementation is explained below:

1. *Extract features from Profile Database and calculate the Profile Score* - First a map is used and initializes the profile scores for all of the users. Based on search keyword and experience of the users the scores are calculated for all the users who are related to a particular field / fields.

```
Map<String, Double> profileScore = new HashMap<String,
                                                Double>();
ResultSet r = s.executeQuery("SELECT * FROM " +
                                                "profiledb");
while(r.next())
{
  String areas = r.getString("area");
  String experience = r.getString("experience");
  String user = r.getString("user_id");
  Set<String> area = getCommaSeparatedValues(areas);
   for(String a : area)
            {
            if(a.equals(id))
            {
            if(profileScore.containsKey(user))
            {
            double score = profileScore.get(user);
            score += Double.parseDouble(experience);
            profileScore.remove(user);
            profileScore.put(user, score);
                              }
                    }
            }
}
```

2. *Extract features from Group Database and calculate the Group Score*– First a map is used and initializes the group scores for all of the users. Based on search keyword and number of groups related to search parameter the scores are calculated for all the users who are related to a particular field and interested in the related groups.

```
Map<String, Double> groupScore = new HashMap<String,
                                                Double>();

String group_ids = r.getString("group_ids");
String user = r.getString("user_id");
Set<String> groupIds =
                    getCommaSeparatedValues(group_ids);
for(String groupId : groupIds)
{
 Statement s1 = conn.createStatement();
 ResultSet r1 = s1.executeQuery("SELECT * from groupdb
                              where group_id = " + groupId);
 while(r1.next())
 {
  String description_keywords =
                    r1.getString("description_keywords");
 String group_score = r1.getString("group_score");
 Set<String> keywords =
        getCommaSeparatedValues(description_keywords);
 for(String key : keywords)
   {
   if(key.equals(id))
    {
    if(groupScore.containsKey(user))
     {
      double score = groupScore.get(user);
       score += groupScore.remove(user);
       groupScore.put(user, score);
       }
     }
      }
    }
}
```

3. *Extract features from Communication Database and calculate the Communication Score* – A map is used and initializes the communication database scores for all of the users. Based on search keyword and number of communication threads related to search parameter the scores are calculated for all the users who are related to a particular field.

```
Map<String, Double> communicationScore =
                              new HashMap<String, Double>();

String id = getWordIdFromVocabulary(K);

String discussionWords = r.getString("discussion_word");
String user = r.getString("from_user");
Set<String> keyWords =
getCommaSeparatedValues(discussionWords);

for(String word : keyWords)
{
  if(word.equals(id))
    {
     if(communicationScore.containsKey(user))
      {
          double score = communicationScore.get(user);
          score += 1.0;

          communicationScore.remove(user);
          communicationScore.put(user, score);
          }
      }
 }
```

4. *Calculate the Aggregate Score of a user by combining individual scores* - Once the scores from all the social network databases are calculated then the next important step is to combine these individual scores and calculate the aggregate scores of the users. These aggregate scores will be used to generate the ranking of the users for PRTF protocol.

```
Map<String, Double> communicationScore =
                    CalculateScores.getCommunicationScore(K);
Map<String, Double> groupScore =
                    CalculateScores.getGroupScore(K);
Map<String, Double> profileScore =
                    CalculateScores.getProfileScore(K);

Set<String> users =
          CalculateScores.getUsers(communicationScore,
                                groupScore, profileScore);

Map<String, Double> aggregateScore =
                              new HashMap<String, Double>();

for(String user : users)
{
```

```
Double totalScore = 0.0;
if(communicationScore.keySet().contains(user))
        {
        totalScore += communicationScore.get(user);
        }
        if(groupScore.keySet().contains(user))
        {
                totalScore += groupScore.get(user);
        }

        if(profileScore.keySet().contains(user))
        {
                totalScore += profileScore.get(user);
        }
        aggregateScore.put(user, totalScore);
 }
```

5. *Generate Ranks of the user from the Aggregate Scores of the users*– After aggregate score calculation for all the users in step 4, the aggregate scores are sorted and top users are selected as the result to the given query which is list of users selected for a particular field. In this manners the ranked list of the users are generated for the particular field.

```
Set<String> filteredUsers = new HashSet<String>();
if(scoreList.size() > MAX_USERS)
{
   for(int i = 0; i<MAX_USERS;i++)
     {
      for(String user : aggregateScore.keySet())
       {
          if(aggregateScore.get(user) == scoreList.get(i))
            {
                filteredUsers.add(user);
             }
      }
   }
}
```

VI. EXPERIMENTAL ANALYSIS

After implementing the PRTF algorithm a test simulation was run over the social networking databases as mentioned in the CM university social networking data [11]. The sample data snapshot taken for illustrative example are given below for various databases. Based upon the sample data taken the following is the aggregate scores for all the users for the search of field “Java”

D :: 13.5
A :: 35.25
C :: 43.0
B :: 5.0
E :: 14.5

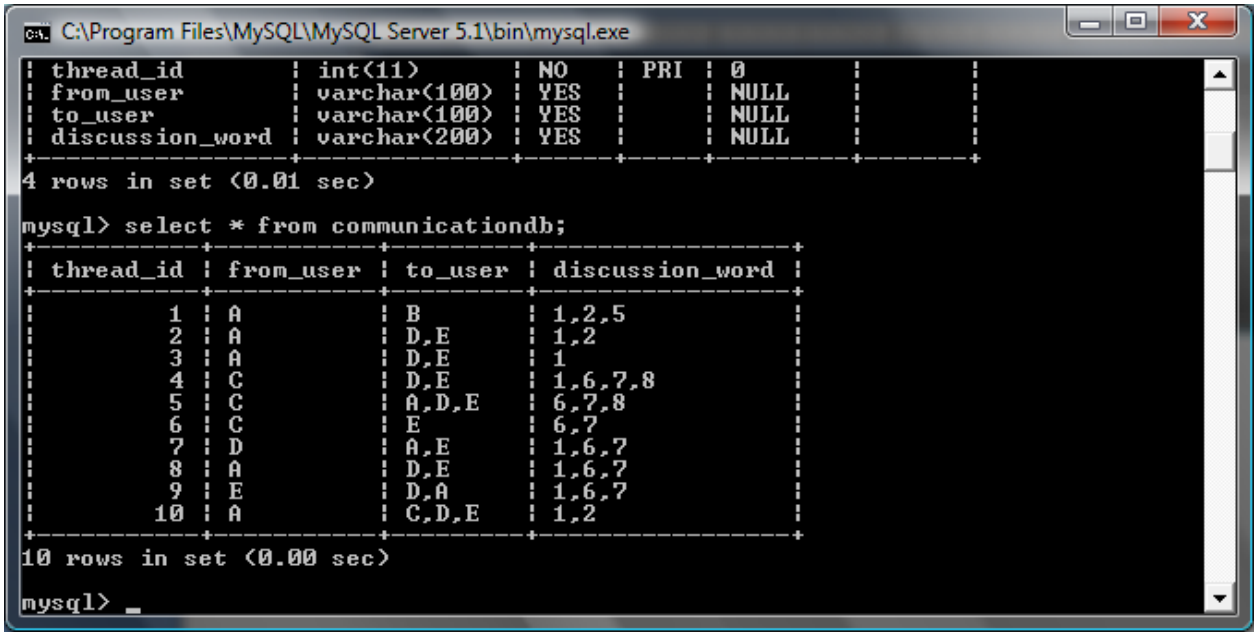


Fig1. Fields and Values available in communication db

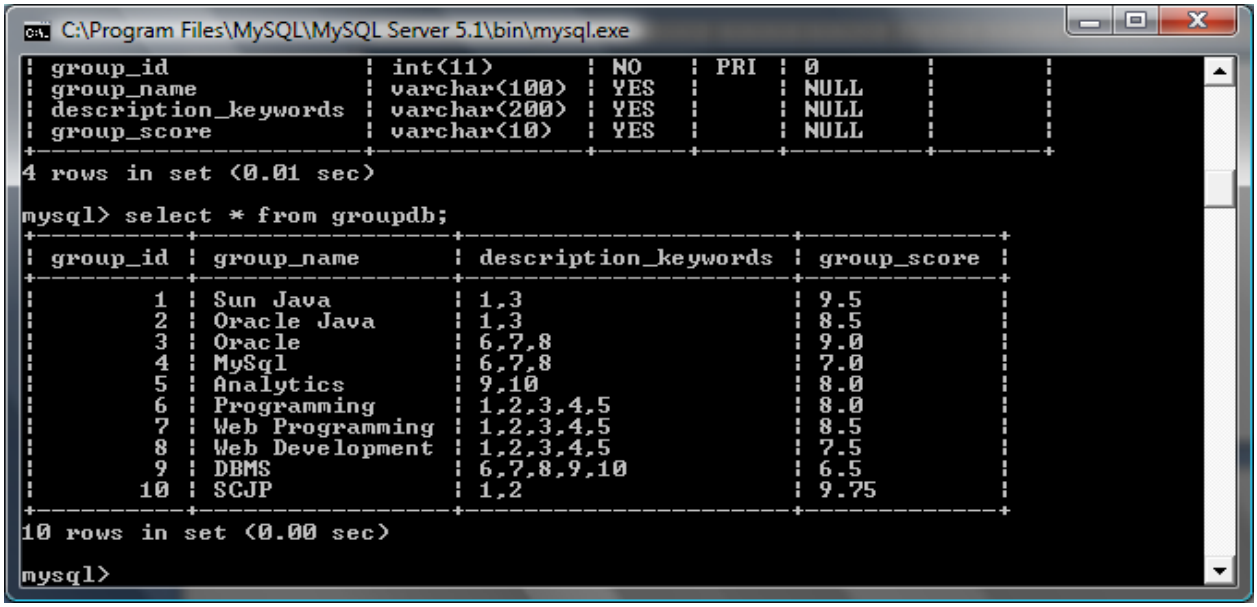


Fig 2. Fields and Values present in groupdb

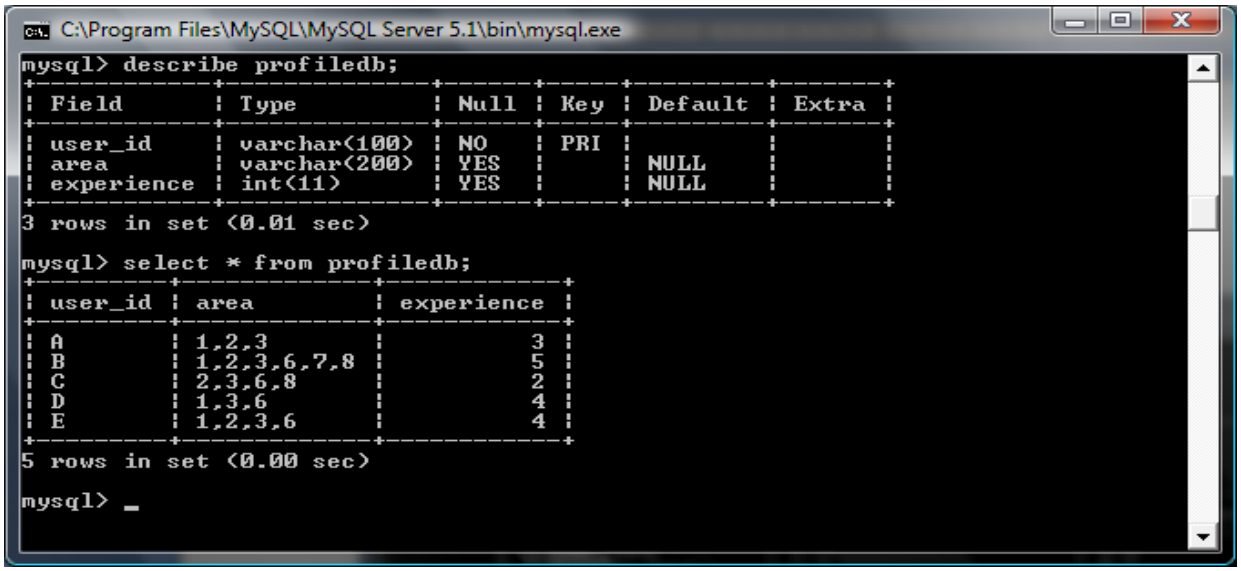


Fig3 Structure of profiledb and data present for testing

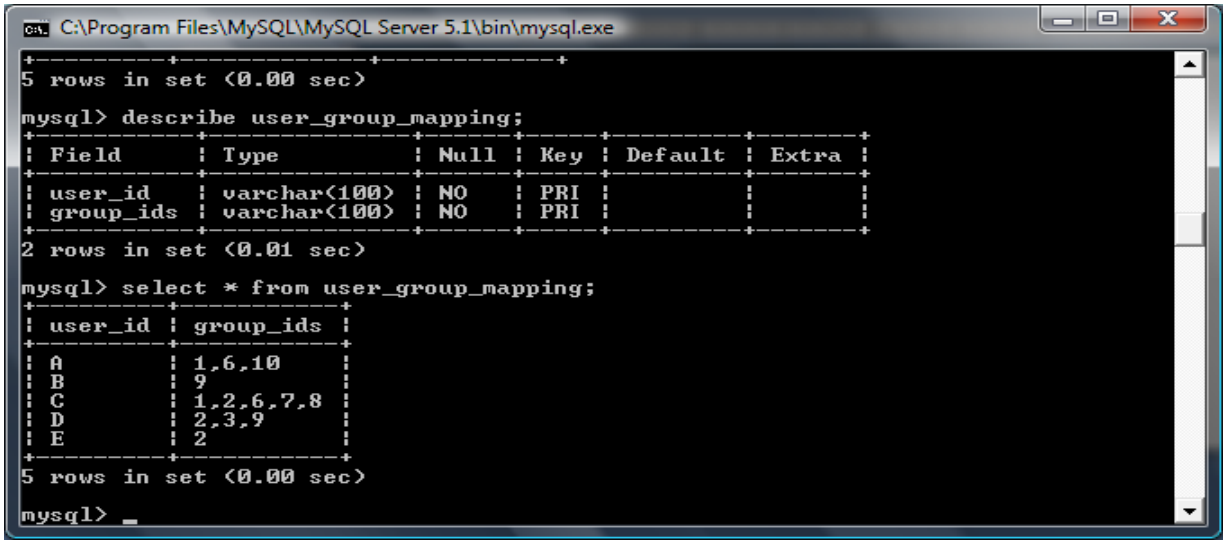


Fig 4 User Group Mapping Structure and Data

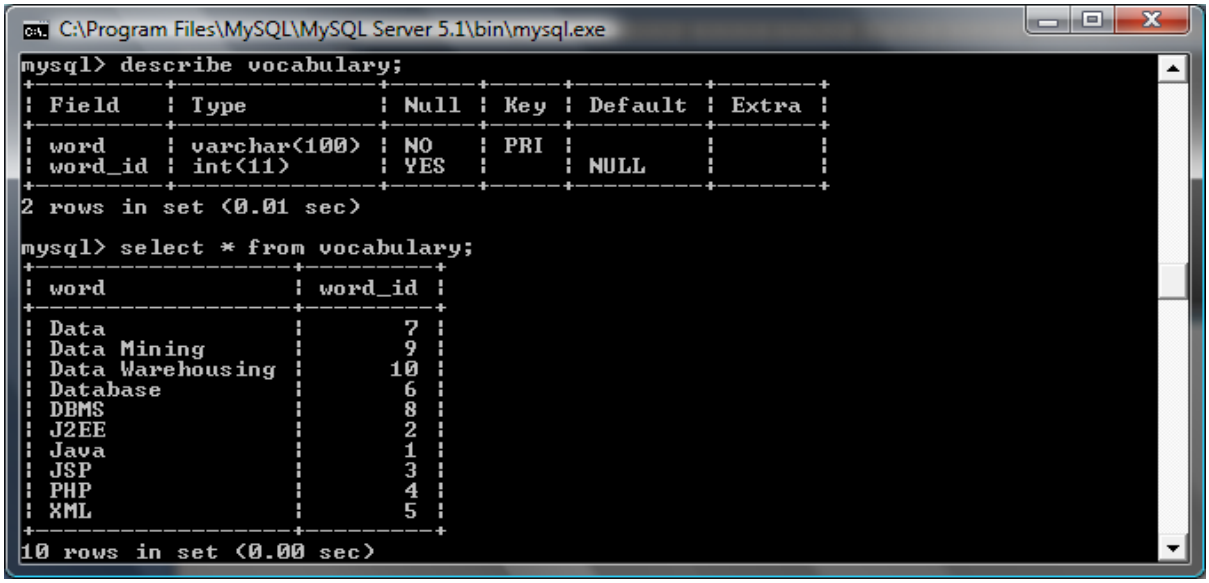


Fig 5 Vocabulary Structure and Data

VII. OUTPUT

The following is the ranked list of the users appearing as the result of PRTF for the search keyword "Java". Top three users have been returned as the short listed experts as per our requirement.

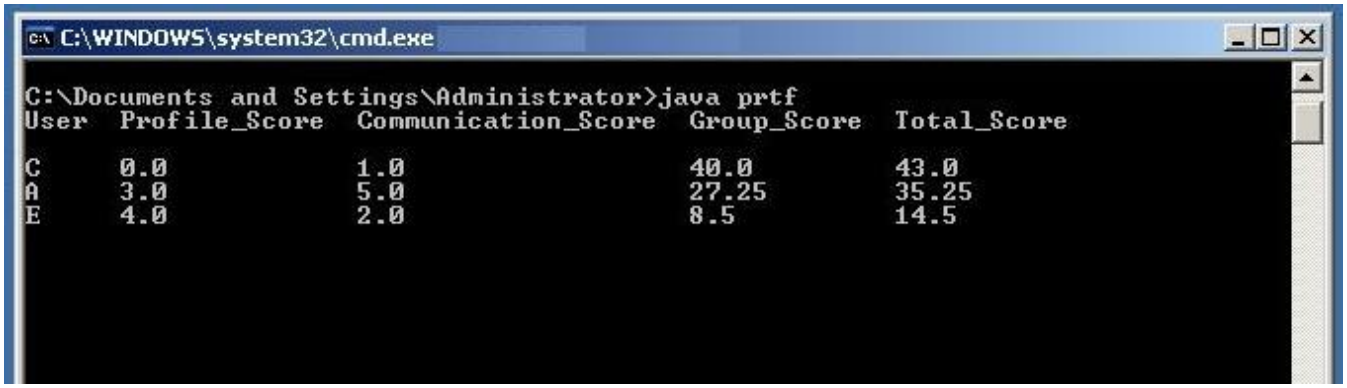


Fig 6. PRTF Output – Ranked list of experts

VII. CONCLUSION

The PRTF algorithm was proposed in [1]. In this paper an open source framework pertaining to the PRTF is implemented using the popular open source technologies like Java, MySql etc. There could be number of applications based on PRTF and so the implementation gives a new dimension to the social network mining. Following are some of the typical scenarios where the proposed framework to implement PRTF can support social networking mining:

A. The proposed framework can be used for finding an experts on a topic / subject. If the field (search) keyword is from any technical area then it can be seen as the technique to find an expert in the particular technical area. This method of expert finding is different from other methods, since the searching of experts is done across all the segments of the social networking databases. Finally the search result data is integrated and ranked as per the requirement.

B. The proposed framework can also be used for creating new groups and new forums on the basis of some search fields. For example suppose some one is searching for the field "Corruption", then the set of persons retrieved as the outcome of PRTF protocol can be used to form new group called "Corruption Group" or a forum can be created like "Corruption Forum" where the persons retrieved can act as participant members and person with top rank can act as the moderator / administrators etc.

REFERENCES

[1] Ashok Bhansali and Dr. HR Sharma, "PRTF: Person Related to a Field Protocol for Searching in Social Network Databases", Journal of Global Research in Computer Science, Pages: 21-26, Dec' 2010.

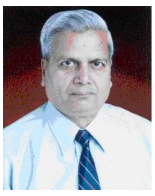
[2] I-Hsien Ting , "Web Mining Techniques for On-line Social Networks Analysis", Service Systems and Service Management, 2008, Page(s): 1 - 5, 2008

[3] Duncan J. Watts, Peter Sheridan Dodds, M. E. J. Newman, "Identity and Search in Social Networks", Vol. 296.

no. 5571, pp. 1302 – 1305, Science 17 May 2002
[4] Nan Du, Bin Wu, Xin Pei, Bai Wang and Liutong Xu, "Community Detection in Large-Scale Social Networks", International Conference on Knowledge Discovery and Data Mining archive, Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis table of contents, San Jose, California , Pages: 16-25, 2007
[5] Dawit Yimam, "Expert Finding Systems for Organizations: Domain Analysis and the DEMOIR Approach", ECSCW 99 Beyond Knowledge Management: Management Expertise Workshop, 2000
[6] Krisztian Balog Maarten de Rijke, "Finding Experts and their Details in Email Corpora", In International World Wide Web Conference, Proceedings of the 15th international conference on World Wide Web. Pages, 1035-1036, ISBN 1595933239, 2006
[7] Krisztian Balog and Maarten de Rijke, "Determining Expert Profiles (With an Application to Expert Finding)", International Joint Conference On Artificial Intelligence archive, Proceedings of the 20th international joint conference on Artifical intelligence, Pages: 2657-2662, 2007.
[8] Christian Bird, Alex Gourley, Prem Devanbu, Michael Gertz, Anand Swaminathan, "Mining Email Social Networks", MSR'06, May 22–23, 2006, Shanghai, China
[9] John G. Breslin, Uldis Bojars, Boanerges Aleman-Meza, Harold Boley, Malgorzata Mochol, Lyndon JB Nixon, Axel Polleres, and Anna V. Zhdanova, "Finding experts using Internet-based discussions in online communities and associated social networks", In Proceedings of the 1st International ExpertFinder Workshop Workshop at Knowledge Web General Assembly 2007, 2007.
[10] Jian Jiao, Jun Yan, Haibei Zhao, Weiguo Fan, ExpertRank: An Expert User Ranking Algorithm in Online Communities, 2009 International Conference on New Trends in Information and Service Science, pp. 674 - 679.
[11] CM University Data - http://www.cs.cmu.edu/~awm/10701/project/data.html

SHORT BIODATA OF THE AUTHORS

Ashok Bhansali completed his graduation from NIT-Jamshedpur and then pursued his M.Tech. in Computer Technology from NIT-Raipur. He is a SUN certified Java programmer and has more than 15 years of industrial & academic experience. He has served in various reputed organizations like Nelco, TechMahindra, SSCET etc. He is a member of ISTE, IEI, CSI and has published many research papers in international journals and conferences. He has been the chairman of various bodies and organized many conferences and STTP. Presently he is serving the OP Jindal Institute of Technology as Associate Professor in the Department of Computer Science and Engineering.



Dr. H.R. Sharma did his M.Tech Computer from Delhi University and completed his Ph.D. from IIT Delhi in Computational Mathematics. He is having more then 38 years of academic experience. He has been the chairman and member of many government and autonomous bodies. He has guided many research scholars and his research area includes Computational Mathematics, Analysis of Algorithms, AI & ES, and NLP. Presently he is working as Director CSITS Durg (C.G.). He is a member of ISTE, CSI, and IEEE.