2009

Expert Finding Using Social Networking

Parin Shah
San Jose State University

EXPERT FINDING USING SOCIAL NETWORKING

A Writing Project

Presented to

The Faculty of the Department of Computer Science

San Jose State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Parin Shah

Fall 2009

ABSTRACT

In today's world, knowledge transfer is considered an important and essential activity for the success of an enterprise. Large corporations have realized the need to reuse existing knowledge rather than spend time and effort on solving the same problems again. For these reasons, most corporations now have knowledge repositories. These repositories are visited for possible solutions whenever there is a problem that cannot be easily resolved by using the expertise of the existing team. Apart from this, the problems faced by the people in the company can also be resolved by asking for help from expert in that problem domain. This approach proves to be more efficient in terms of time and manual efforts, while also saving resources. This report proposes a strategy of finding an expert in a required domain by analyzing a company's social network i.e. communication amongst its employees. Efficiently finding an expert is one of the most important tasks currently faced by the information industry and this problem has not been sufficiently addressed in the past. Conducting further research in this field can improve the time required to solve critical time-sensitive problems in an enterprise environment, thereby improving the overall efficiency of the enterprise.

Keywords: Social, networking, experts, expert systems

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Robert Chun, whose guidance, support, and dedication is invaluable. Dr. Chun is an educator in the truest sense of word. I also appreciate the participation of Dr. Agustin Araya and Dr. Mark Stamp as thesis committee members. The committee has provided me with valuable insight and my work presented in this report is deeply influenced by their guidance.

I would like to thank Prof. Janelle Melvin, who taught me to write the Master's thesis report in a correct and professional manner.

It has been a challenging journey that would have been difficult to complete without your ongoing support.

Thank you.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Social Network is a collection of likeminded people in close proximity to each other. This proximity can be physical or virtual. The main use of social networks includes helping people within the network, sharing knowledge in a particular area of interest or having a casual interaction. A social network may consist of experts in a specific domain and other people who join the network to receive help from these experts. These social networks can be used to search for an expert within a group of people.

Searching an organization's repository to find an expert is a frequently faced challenge. Searching for problem solving information on issues that have already been resolved in the past is like reinventing the wheel. The complex problems faced in today's world requires both an increased amount of labor as well as it incurs additional cost. Both these factors are detrimental to the interest of any growing corporation. "Knowledge workers still spend about 15% to 35% of their time searching for information and 40% of the corporate users can't find the information they need on intranets" (Susan Feldman, 2004, p.8). These numbers are not very motivating; hence there is a need to find an expert in a particular domain. Also, finding an expert should be easy and time efficient.

While exploring this domain, we found that expert search using social network is very different compared to finding a document using a search engine. This is due to the fact that unlike documents there is no text or keywords associated with the experts. To find an expert using the documents in the archive, an expert has to be associated with some keywords that best describes him. To get better results, the keyword description should be as close as possible to the real description. People within a social network have an explicit relationship with other people in the group. Once the keyword description document of every person in the network is completed, these relationships can then be used to find the experts in a specific domain. The extended definition will be covered in section 2. Section 3 consists of the hypothesis in the present scenario followed by an algorithm that is devised for solving the problem of expert finding in section 4. The results proving the competitiveness of the new link weight approach compared to the existing personal profiling approach are illustrated in section 5. The paper is concluded in section 6 and the future possibility on the research topic is enlisted in section 7.

2. EXTENDED DEFINITION

Social Network is a collection of likeminded people in close proximity of each other. The main uses of social networks include helping people within the network, sharing knowledge in a particular area of interest or having a casual interaction. These social networks may also contain experts in their respective domain. The proximity of the participants in the social network can be of following types:

- **Physical Proximity**: In this case, people within a social network meet in person and resolve problems faced by them and other members of the network. A yoga group, a particular fraternity in school or a math club in a computer science department are good examples of social networks with physical proximity.
- **Virtual Proximity:** In this case, technology is used to simulate the effect of a meeting and there is no face to face interaction amongst the members. Computer networks are used to communicate their ideas and resolve any issues. Facebook, LinkedIn or Blogs are the examples of social networks with virtual proximity.
- **Hybrid Proximity:** These social networks use technology for their regular interaction and people within the network also have direct face to face interaction. IEEE and other groups are examples of such proximity within social networks. People within these groups have regular interaction using emails or blog facilities provided by IEEE. They also meet in person for events such as a conference.

How are social networks formed? Social networks start with some likeminded people who are interested in sharing their ideas or collaborating on a particular research. These people are pioneers and they form a group or a community. Once other likeminded people come to know about the existence of a group or community, they come forward and join it.

The people within this particular community think they can improve the effectiveness and efficiency of their work by means of collaborating with other people who have similar thoughts. This method proves to be productive i.e. consider an example of a scientific community. If there is no sharing of information among the researchers in a similar domain, a lot of research would be repeated due to the lack of knowledge about its existence. By means of social networks like IEEE conferences, these researchers come together and publish their work. Their research is available to other scientist so that they too have knowledge of advances happening in their domain.

How do we find an expert using a social network? As mentioned earlier, people in a social network have an explicit relationship with other people within the group. So, there may be a case where a person is connected to 100 people and there can also be a case where a person may be connected to one or two persons. Fig 1 gives a graphical representation of a typical social network. There are three dimensions that can be clearly seen in this figure. The first dimension is a node. Every person in the social network is represented by a node in Fig 1. The second dimension is the size. The size of a node denotes the influence of the person in their specific domain. The third dimension is the link. The link between the nodes denotes a direct connection between two people.

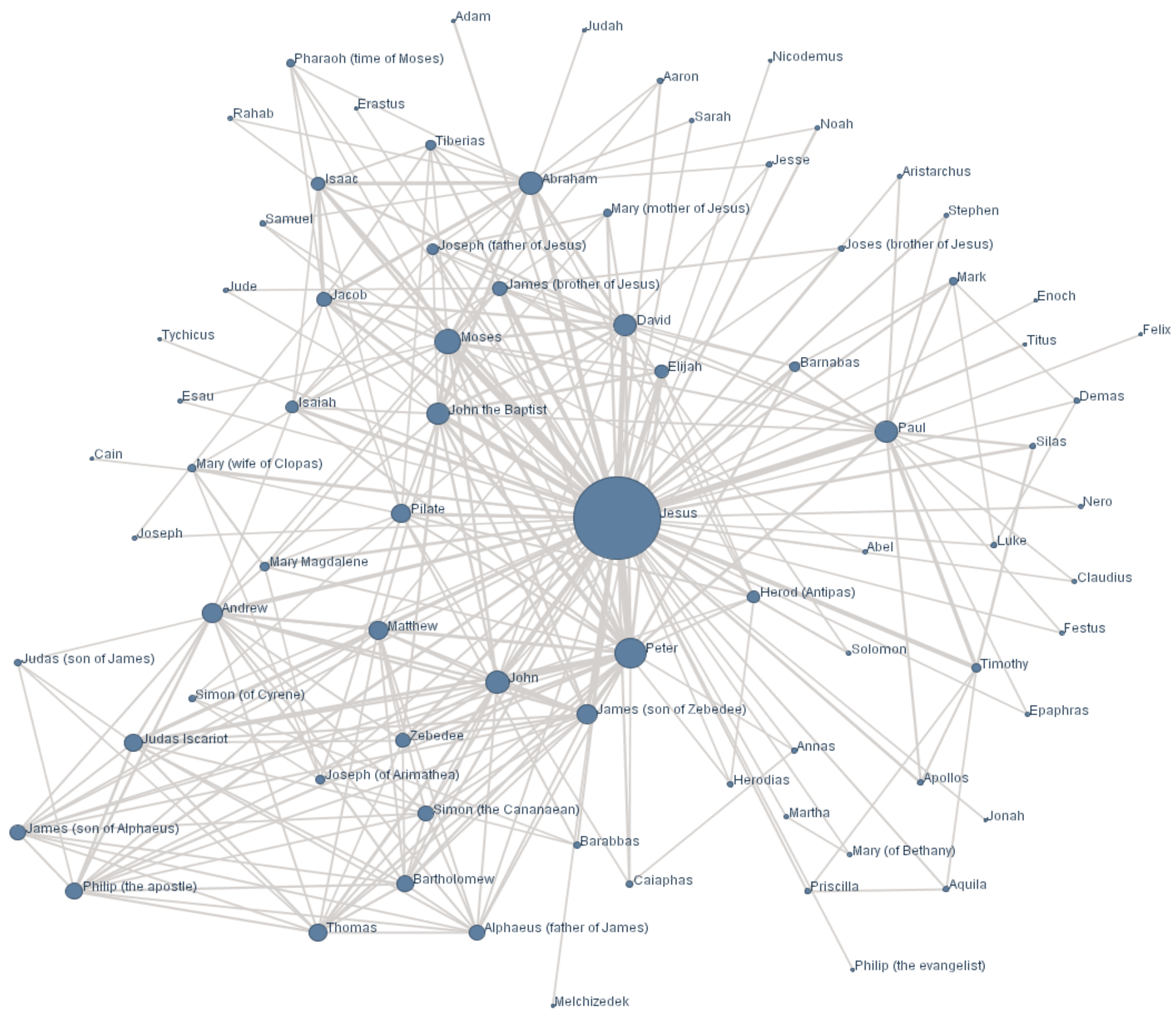


Figure 1: Graphical representation of a social network

3. RELATED WORK

Expert finding is a relatively new field of research which has a lot of scope. In recent times, a lot of researchers have put considerable efforts in this area. They have proposed different approaches to find an expert on the intranet. While some of them are still considered for research interest [8][11], others are developed into products of commercial use. One of the initial approaches to expert finding consisted of manually entering the expert information into the database [1]. Once this is done there exists an expert profile in the organization's database. Later, when the user wants to get information about an expert having knowledge in the required field, he can search the expert profiles in the organization's database. The problem with this approach was that it required manual efforts to keep the expert profile up-to-date. Studies have shown that even if efforts are periodically put into updating the expert profile, they do not provide efficient and accurate results. One of the reasons for this shortcoming is that the people who are entrusted the task of maintaining the expert profile do not have precise knowledge of the experts. And the experts usually find themselves busy to perform this task. Also, manually exploiting information of every individual in an organization was extremely laborious and costly. Thus, the manual entry of expert profiles into the organization's databases did not prove to be a feasible strategy.

Although the technique of finding an expert using manually built expert profile did not provide required accuracy and efficiency, it did have the very approach which later on came to be known as "Expert Profiling"[1][8][11]. Currently two types of approaches are used to find an expert using social network analysis. One of the approaches called "Personal Profiling" has been there for quite some time and the other approach known as "Document Profiling" is relatively recent. Either of these profiling is used in the research that has taken place till date. Both these approaches have their advantages and disadvantages when it comes to a specific application. The approach used also depends upon the structure of the organization, the hierarchy of the organization and the mode of communication used.

In Personal Profiling [1][8][11], the description of the person is simulated by a document that contains the keywords related to that person. This document description should be as close to the actual description of the person and the success of the entire algorithm depend upon it. Once the keyword description document is ready, it can be used to find the expert in the required domain. When the user inputs the keywords for a specific domain in which an expert is to be searched, the search engine matches the user input with the document description of each person. The output of this algorithm is a list of people who are related to the required domain. The major part of this process in performed offline and hence this approach is more efficient compared to others.

In Document Profiling [13][14][15][16], the user input is run on the entire set of actual documents. The output of this phase is a set of documents that are related to the input query. Once the set of related documents is obtained, these documents are further analyzed and the names of the experts are then extracted based on their occurrences in the selected set of documents. This is a runtime process and usually takes longer to find the expert but it proves to be more accurate as the actual documents are being tested. Also, every time the experts are requested, the implementations using this approach use the latest set of documents as their input.

The author in [8] had a very different approach. Although, the baseline algorithm used the personal profiling approach, the author had devised the algorithm to be as automated as possible. The implementation was agent based. Every user of this implementation has a user agent installed on his computer. The user agent is responsible for creating the expert profile of the host. The user agent does not provide the expert advice but it directs us to the required expert. When the user agent is requested for an expert in a specific domain, it goes on and searches other experts who are logged in and have necessary knowledge. This algorithm was implemented for the JAVA language community. Such an approach might work well with a community which has a well defined and limited set of words in their language, but when it comes to the English language it would be difficult to obtain similar accuracy. Also, this approach requires the user to be logged in the system or activate its user agent to participate in the community. Apart from the privacy issues, there is also a requirement for some centralized server that keeps a track of the users that are logged in.

The authors [1] used the approach of expert profiling. Their algorithm consists of expert profiling as a major part followed by the usage of the ranking algorithm. This algorithm aims at using the person centric approach to find the expert rather than using the keyword retrieval methods on documents. The person's profile is generated using web pages and the tags in the HTML language are used to assign weights to the text and find the keywords to generate the CDD (Candidate Description Document). Once the CDD for all candidates is in place, a keyword search is performed for the required expert domain on these Candidate Description Documents in the database. The output of this search is a list of probable experts. This list is unsorted and is given as an input to the ranking algorithm. The ranking algorithm gives the ranked list of experts in the required domain as the output. One of the major drawbacks of this approach is that it concentrates on web pages as the primary source of information for profile building. There are other more important modes of communication such as chat logs and emails that can be used to build the expert profile more accurately, thereby providing with more accurate results.

Paper [11] talks about the agent based approach using person profiling. This algorithm consists of a user contact file which consists of the users' contacts and the keywords associated with each of the contact. The user agent also consists of a list of keywords that are retrieved from the user documents. Once this is ready, Netscape browser is used as a front end to query the system. When a query is asked, the user agent checks for the required keywords in the user contact file, if a match is found, the search stops and returns the results. If no match is found then the user agent propagates the search query to the user agent of the contacts and the contacts user agent performs a similar search. This continues until all contacts are exhausted or a threshold is reached. The main concern with this approach is dealing with privacy. Also, there is a requirement of a user agent on the user machine and the user has to be logged in to use the system. The value of the threshold is also a determining factor. If the value is set very low then the query may not provide any result and if the value is set very high then there is a possibility of wasting time after which there may be no results.

The algorithm [12] generates a unique document per user. First, a query with the employee name is entered into the system. The output of this query is a list of all the documents that have the name of the employee associated with them. The text of all these documents is concatenated and put into a single file. This file is then used to find the experts and rank them. Once this unique document is created for each employee, the second phase of the algorithm starts. The first phase is an offline process. The second phase starts with querying the database with the required keywords. These keywords are related to the required experts. When this query is executed, it checks the concatenated document of each person.

The ranking is done according to the occurrence of the keywords in the person's document. In this approach, the entire file is checked for the keywords and there can be a number of copies of the same file. This implies that the actual number of documents searched is more than the number of documents present. Thus, this approach seems to be more time consuming.

The algorithm [13] takes the document profiling approach. Here, the documents are divided into three categories: papers written by the candidate, papers in which there is a mention of the candidate and papers in which there is no mention on the candidate. The documents written by the candidate and documents in which the candidates are mentioned are taken into consideration. Different weights are assigned to the candidates' occurrences in both the types of documents. The candidates are ranked based on the occurrences in both types of documents. A unique formula is designed that takes into consideration the occurrences and the weights, to assign ranks to the candidates. This processing is performed online and hence it takes a considerable amount of time to provide results.

In paper [14] has an approach similar to paper [13], but a special consideration is given to the resume and a heavy weight is assigned to this document. Due to the usage of resume in finding out the expert, the accuracy of this algorithm is greater than the previous mentioned similar algorithm. This processing is done online and hence it takes a significant amount of time to provide the results.

The author of [15] follows the document profiling approach. First, all the documents in the repositories are queried for a given topic. The output of this query is a list of all the emails that are related to this topic. Once this is done, all the people involved in the retrieved documents are listed. A graph is created from the relationship among these people. Once the graph is built, a modified HITS algorithm is used to rank the candidates. The output of this method is a ranked list of experts on the requested topic. The only drawback of this algorithm is that every time a query is submitted the entire process is repeated and hence this process is time consuming.

The approach [16] is completely based on document profiling. The information for this algorithm is collected from the internet and this information is used to create a social network. This approach was designed for a number of purposes such as searching for a given person's information, a publication, an expert on a given topic and associations between researchers. The co-authorship is used as a building block for the relationship in this approach. A social network is built after identifying the related web pages. Once this is done, the candidate's information is separated from these documents. Then a social network is constructed from the information obtained. Scores are assigned to each of the candidate and these scores are then propagated amongst the candidates. This score is then used to rank the experts. The drawback with this approach is that it was based on World Wide Web, so the accuracy was not very important when this approach was being designed. Their primary concern was to make a system that is more flexible but they compromised on accuracy.

4. ALGORITHM

This section describes in detail the new link weight approach that is used to improve the results while using social networks for ranking the experts. There are 4 phases in the newly developed algorithm to find an expert using social network analysis. These phases are:

1. Building a Communication matrix
2. Generating Keyword description document of an individual
3. Finding probable experts
4. Using communication matrix to rank the experts

4.1 Building a communication matrix

The first phase consists of analyzing the entire repository of all the email communication among the people in an enterprise. The virtual representation of this communication matrix can be viewed as a two dimensional array. The rows in the 2-D array contain the strength of the communication between the person in the first column of that row and the person that belongs to any other column of that row. The communication matrix thus gives a complete representation of the communication that takes place in the enterprise. This is helpful to find the relationship amongst people in the enterprise and also the strength of their relationship. Fig 1 shows the graphical representation of an email network. The links between the nodes denote the email communications and the thickness of these links denote the strength of their relationship. The communication matrix built in phase 1 is later used in phase 4 for ranking the experts on a particular query input. The data structure that can be used to represent this phase is a two dimensional array. The size of this array is of the order N x N, where N represents the number of people in the enterprise. The value of N can be large, hence for efficient storage of the communication matrix; research is on-going to find a new data structure. This activity of building a communication matrix is an offline and cyclic task and can be done every week, every fortnight or as per business requirements.

4.1.1 Link weight algorithm

A custom algorithm is used to construct a communication matrix in phase 1. This algorithm when used in conjunction with the existing expert ranking techniques would improve the quality of their results. The use of the Link weight algorithm would provide the results in a manner such that greater weight is given to those experts during the ranking process whose probability of responding to the query is sooner than the others. The user would get a list of experts that would be ranked taking into consideration the probability of the expert replying to the user query. Thereby, the probability of the user getting a quicker response increases.

The link weight algorithm is used in two parts during the process of finding an expert. During part 1, it is used to create the communication matrix which is an offline process. The second part of the link weight algorithm is executed in real time and is used to rank the experts in phase 4.

4.1.1.1 Link Weight algorithm Part-1

Every email document in the archive is processed by the preprocessor module to create a communication matrix. In the email communication, there is a SENDER and a RECEIVER. In some of the emails there is a possibility of a CC. It is our assumption that the RECEIVER is considered as the most important entity involved in the email conversation followed by CC and then the SENDER. When we preprocess an email, this information is used in the first part of the link weight algorithm.

{SENDER}→ {RECEIVER}

The above notation is used to show the weight of the relation between the sender and the receiver from the sender's side.

{SENDER}→ {CC}

Similarly the above notation is used to show the weight of the relation between the sender and the cc from the sender's side.

The process of creation of the communication matrix is explained using an example. Give appropriate weights to the SENDER, RECEIVER and CC. According to the algorithm, the receiver is given the highest importance during the communication, and hence is assigned a weight of 1. All other weights are given relative to the receiver.

Let's consider following weights for demonstration purpose:

RECEIVER = 1

SENDER = 0.1

CC = 0.5

Email-1

To: tom@yahoo.com

Cc: peter@yahoo.com

Sender: mike@yahoo.com

Hi Tom,

Please get me this work done.

Regards,

Mike.

Using the part-1 of the link weight method to create communication matrix for email-1

{mike}→{tom} = 0.1

{mike}→{peter} = 0.1

{tom}→{mike} = 1

{peter}→{mike} = 0.5

Mike is the sender of the email and Tom is the receiver so the weight between Mike and Tom from the side of Mike is 0.1, while the weight between Mike and Tom from Tom's side is 1. In the same email, Mike is the sender and Peter is cc. So, the weight between Mike and Peter from Mike's side is 0.1 and the weight between Mike and Peter from Peter's side is 0.5.

Email-2

To: mike@yahoo.com

Sender: tom@yahoo.com

Hi Mike,

The work is done and this email is a confirmation.

Regards,

Tom.

Using the same explanation as before, a communication matrix for email-2

{mike}→{tom} = 1

{tom}→{mike} = 0.1

All the emails in the archive would be processed in the above mentioned way. The final entries in the communication matrix would be the addition of the entries from the individual emails. Once the matrix is created for each email, the entries from the matrix of all the emails are added and the preprocessing step is over. This matrix is stored in the memory for use in phase 4. The final entries in the communication matrix for the above two emails would be as follows:

{mike}→{tom} = 1.1

{mike}→{peter} = 0.1

{tom}→{mike} = 1.1

{peter}→{mike} = 0.5

4.1.2 Selection of weights for SENDER, RECEIVER AND CC

There are three entities in the email conversation that are given weights according to the proposed algorithm. The RECEIVER being the most important entity of the conversation is given the weight of 1 and all other weights are given in relation to the RECEIVER. The weights used in the experiments were 0.1, 0.3, 0.5, 0.7 and 0.9 and they were assigned to both the SENDER and the CC. Experiments using all possible weights were performed on three different sets of input keyword phrase. For every keyword phrase, there were 25 variations in the weights of SENDER and CC. For every result, a Drift ratio was calculated. The constant used for comparing the response ratio while finding the Drift ratio is the threshold. Experts below this threshold are mostly either spammers or they are forums. This threshold is configurable and should be customized according the environment where the system is being built. For our case this threshold was 0.2. The rank found using this threshold in the proposed algorithm is compared with the existing technique and the change in the ranks of each expert is found out.

$$\text{Drift ratio} = \frac{\text{Average change in the rank of the expert with response ratio} > 0.2}{\text{Average change in the rank of the expert with response ratio} < 0.2}$$

The drift ratio with a value close to 0 is considered to be good. It means while assigning ranks using the proposed link weight method as opposed to assigning the ranks using the existing personal profiling technique, the experts with the response ratio less than

0.2 are given inferior ranks and the experts with the response ratio greater than 0.2 are assigned superior ranks.

Table 1: Comparison of Drift ratio to find optimal weights for RECEIVER and CC in the proposed link weight technique

Run Number	Receiver's weight	CC's weight	Drift ratio for Database System	Drift ratio for Operating System	Drift ratio for Computer Network
1	0.1	0.1	0.48	0.25	1.09
2	0.1	0.3	0.43	0.21	1.03
3	**0.1**	**0.5**	**0.36**	**0.19**	**1.03**
4	**0.1**	**0.7**	**0.32**	**0.19**	**1.03**
5	**0.1**	**0.9**	**0.26**	**0.19**	**1.03**
6	0.3	0.1	1.08	0.38	1.9
7	0.3	0.3	0.91	0.36	1.41
8	0.3	0.5	0.75	0.39	1.41
9	0.3	0.7	0.63	0.33	1.22
10	0.3	0.9	0.55	0.3	1.14
11	0.5	0.1	2.55	1.9	1.8
12	0.5	0.3	5.54	0.82	1.87
13	0.5	0.5	4.71	0.61	1.62
14	0.5	0.7	3.99	0.52	1.43
15	0.5	0.9	3.77	0.56	1.33
16	0.7	0.1	3.33	2.86	1.71
17	0.7	0.3	2.5	2	1.56
18	0.7	0.5	4.82	1.5	2.11
19	0.7	0.7	5.80	0.8	1.77
20	0.7	0.9	5.89	0.57	1.24
21	0.9	0.1	6.52	1.06	0.94
22	0.9	0.3	8.33	1.39	1.04
23	0.9	0.5	7.26	1.33	1.47
24	0.9	0.7	5.09	2	1.71
25	0.9	0.9	5.80	1.17	1.87

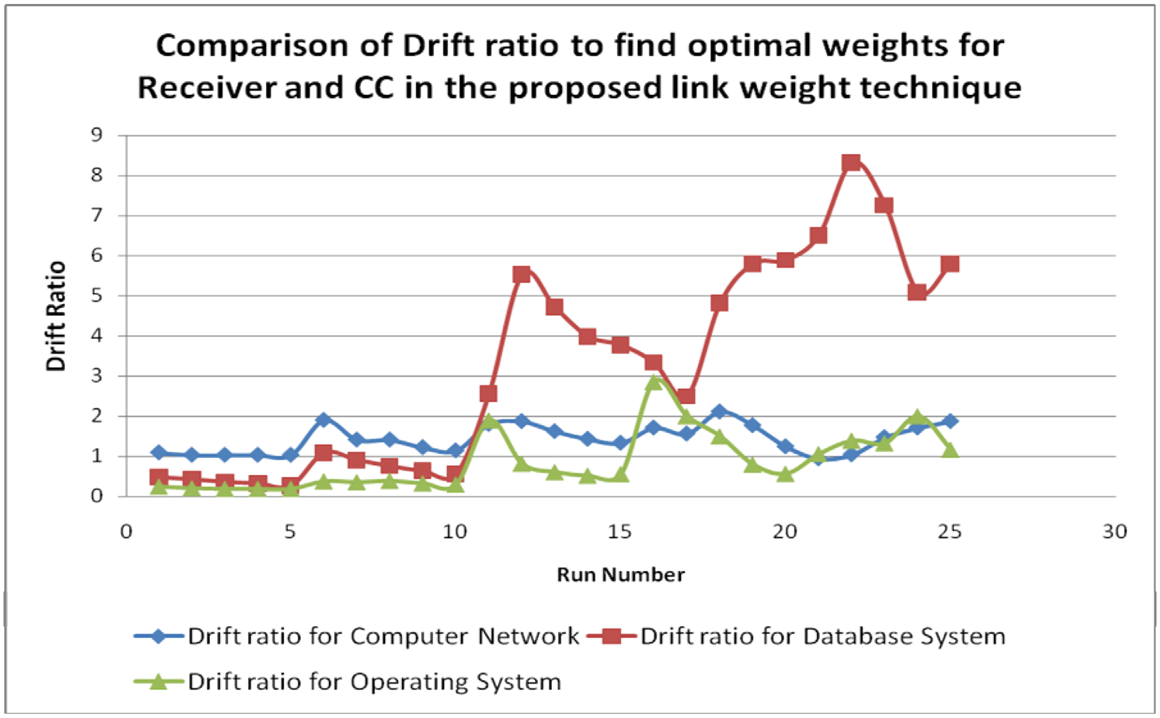


Figure 2: Comparison of Drift ratio to find optimal weights for Receiver and CC in the proposed link weight technique

Table 1 gives the details of each run including the drift ratio. Different weights are assigned to the SENDER and the CC in each run. The drift ratio for all the runs is further used to finalize the weights for the SENDER and the CC. These weights are used to calculate the set of experts for a given keyword search phrase. Analysis of run 3, 4 and 5 from Table 1 and Fig. 2 shows that for all the three search queries, the drift ratio is the least when the SENDER is assigned the weight of 0.1 and the weight of the CC is between 0.3 – 0.7. Thus, we can conclude the SENDER to have a weight of 0.1 and we can average the weight of the CC to be 0.5. For all the calculations henceforth the weights used are as follows:

SENDER : 0.1

RECEIVER : 1

CC : 0.5

4.2 Generating keyword description document

In the second phase all the documents in the archive are processed. This phase is completed offline and has no impact on the execution time. All the documents related to a particular person are concatenated into a single document with the name as the file name. The input given at this stage is a complete repository of the email documents from which the keyword description documents are generated. These documents are then used to find the keywords related to that individual. The frequency based technique is used to find the words that are most important for a person. In this technique, the words with most repetition and significance are taken into consideration. The stop words can be prevented from appearing on the keyword description document of an individual. This can be done by using the domain knowledge of the organization that is implementing the expert search system. These words include ‘is’, ‘or’, ‘the’, etc. Once the keyword association process is complete, a document is generated for the individual. This document is known as the keyword description document and should be as close as possible to the original description of the person in consideration. This process is repeated for every individual in the organization. Fig 3 gives an example of a typical keyword description document for the expert “Parin Shah”. This document consists of all the keywords that describe “Parin Shah” as accurately as possible.

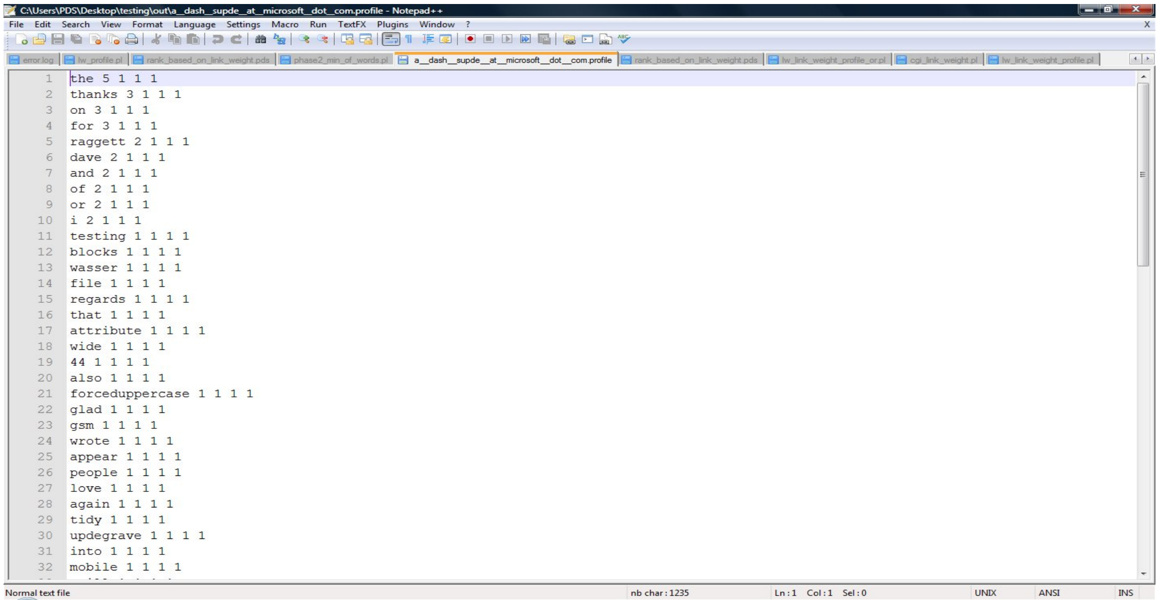


Figure 3: Keyword Description Document for "Parin Shah"

4.3 Finding probable experts

The third phase is executed in real time. In this phase, an efficient algorithm is used to find a list of experts amongst all the people in the network. The input to this phase is given in terms of keywords which are related to the topic in question. This input is then searched amongst all the keyword description documents in the repository. The keyword description documents are generated as the output of phase 2. The keywords in the query are matched against all the keywords in the keyword description document of every individual in the enterprise. Once this is done, we get a list of people who are related to the keywords in the query. This list may be further used to process the experts that have the exact keyword phrase in their email documents. The output for the third phase is a list of experts who specialize in the required domain. This list is unsorted and is of little use to the user. This list is further given as input to the fourth phase.

4.4 Using communication matrix to rank the experts

The fourth phase takes the unsorted list of probable experts from the third phase as input. Now, the part 2 of the link weight algorithm comes into use. The communication matrix constructed during phase 1 is now used. For each probable expert in the list it calculates the following:

Own = summation of {individual expert}→ {other probable experts}

Where, the above expression is the summation of the weight between the individual and all other probable experts from the side of the individual

World = summation of {other probable experts}→ {individual expert}

Where, the above expression is the summation of the weight between the individual and all other probable experts from the side of other probable experts

Response ratio = (Own/ World) if own < world

Response ratio = (World/ Own) if own > world

This ratio gives us the responsiveness of the individual expert to others individuals dealing with similar topic. Here we use a pessimist approach that’s the reason we compute the response ratio using two different formulas. The lower value is taken as the response ratio.

The response ratio is multiplied with the number of emails that contain the given search phrase for every individual to get the final weight for an individual expert. The list of individual experts is further sorted based on this final weight. The output of the fourth phase is a ranked list of experts in the required domain. The list contains the email addresses of the experts that would be displayed in the order of decreasing importance. This ranked list is then presented to the user in the required format.

5. RESULTS

The link weight algorithm was executed a large number of times and the results were tabulated to find out the pattern amongst them. Also, the algorithm parameters were modified and the results after these modifications were tabulated. Once this was performed, a thorough analysis of the results was conducted to find out a range of parameter values for which the proposed algorithm gives the acceptable results.

All the experiments were performed on the Windows pc with the following configuration:

- Intel Core 2 Duo 1.60 GHz processor
- 2 Gb RAM
- 120 Gb Hard disk

5.1 Analysis of the results in terms of efficiency

Experiments were performed to find the efficiency in terms of time for the proposed algorithm. Time efficiency was compared between the existing document profiling technique, existing personal profiling technique and the newly proposed link weight algorithm.

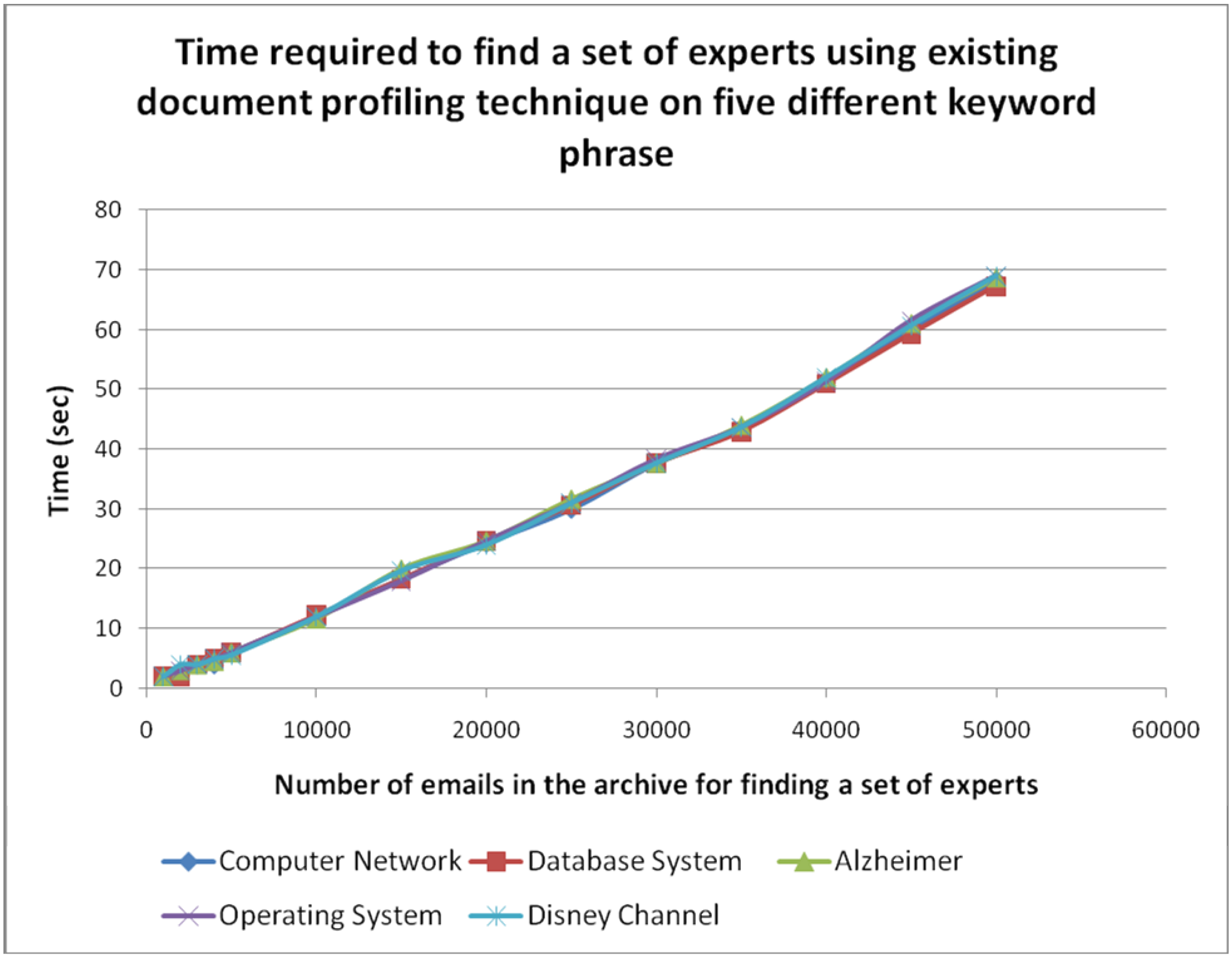


Figure 4: Time required to find a set of experts using existing document profiling technique on five different keyword search phrase

In the document profiling method, there is no preprocessing phase. All the documents in the archive are processed to search for a set of experts in the specific domain. Thus, for each query almost similar amount of time is consumed to find the ranked list of experts. Even if the query returns a negative result it has to search all the documents. Also, from Fig. 4 it can be seen that as the number of documents in the archive increases, the time required to search for a set of experts amongst these documents increases linearly with a higher constant for slope. This linear increase in time to search for experts is detrimental to the efficiency of the search algorithm.

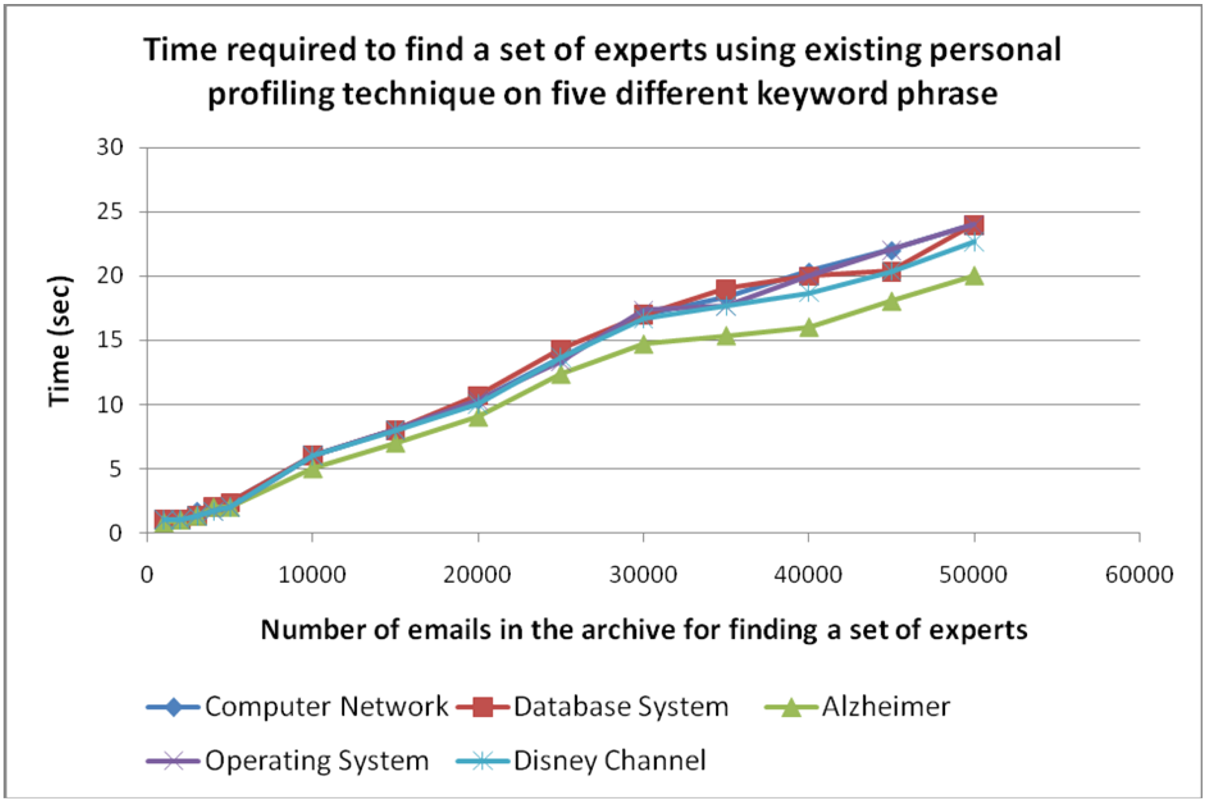


Figure 5: Time required to find a set of experts using existing personal profiling technique on five different keyword search phrase

In personal profiling technique, the documents in the archive are preprocessed. Therefore, the total number of documents to be searched at run time is comparatively very small. This reduction in the number of documents to be searched is reflected by the reduction in the time required to search for an expert using the existing personal profiling technique. Fig. 5 shows that the time required to search an expert using existing personal profiling technique is substantially less compared to the existing document profiling method. The time required to search an expert for Alzheimer is less compared to the other search queries because there were no experts for this keyword so no further processing was required for ranking the experts. The increase in time using this approach is linear, but the slope for this graph is comparatively lesser than the previous document profiling approach.

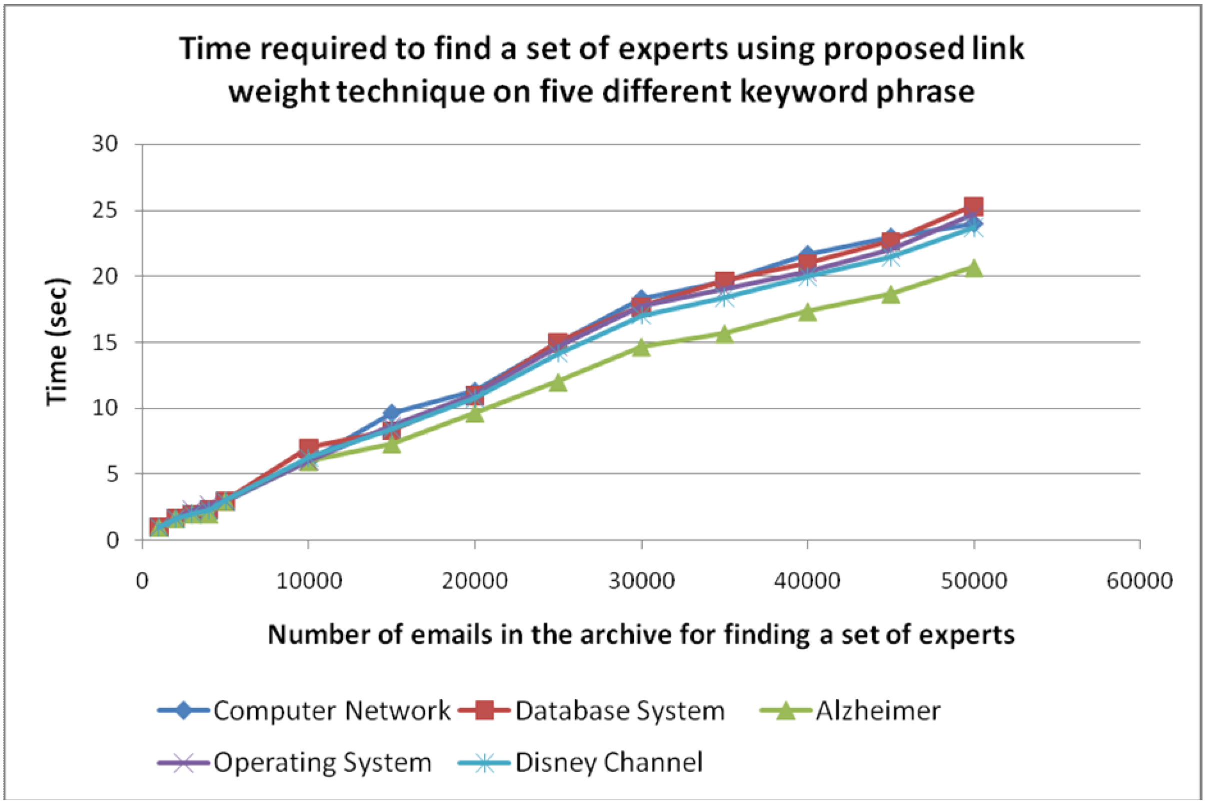


Figure 6: Time required to find a set of experts using proposed link weight technique on five different keyword phrase

In the link weight algorithm technique, the documents in the archive are preprocessed. Therefore, the total number of documents to be searched at run time is comparatively very less. This reduction in the number of documents to be searched is reflected by the reduction in the time required to search for an expert using the proposed link weight algorithm technique. However, the time required by the proposed link weight technique to find the set of experts is slightly greater than the existing personal profiling technique, as more processing is required to be done after the experts are found in order to rearrange them using the response ratio information. Fig. 6 shows that the time required to search an expert using the proposed link weight approach is substantially less compared to the existing document profiling method. The time required to search an expert for Alzheimer is less compared to the other search queries because there were no experts for this keyword so no further processing was required for ranking the experts. The increase in time using this approach is linear, but the slope for this graph is comparatively lesser than the previous document profiling approach.

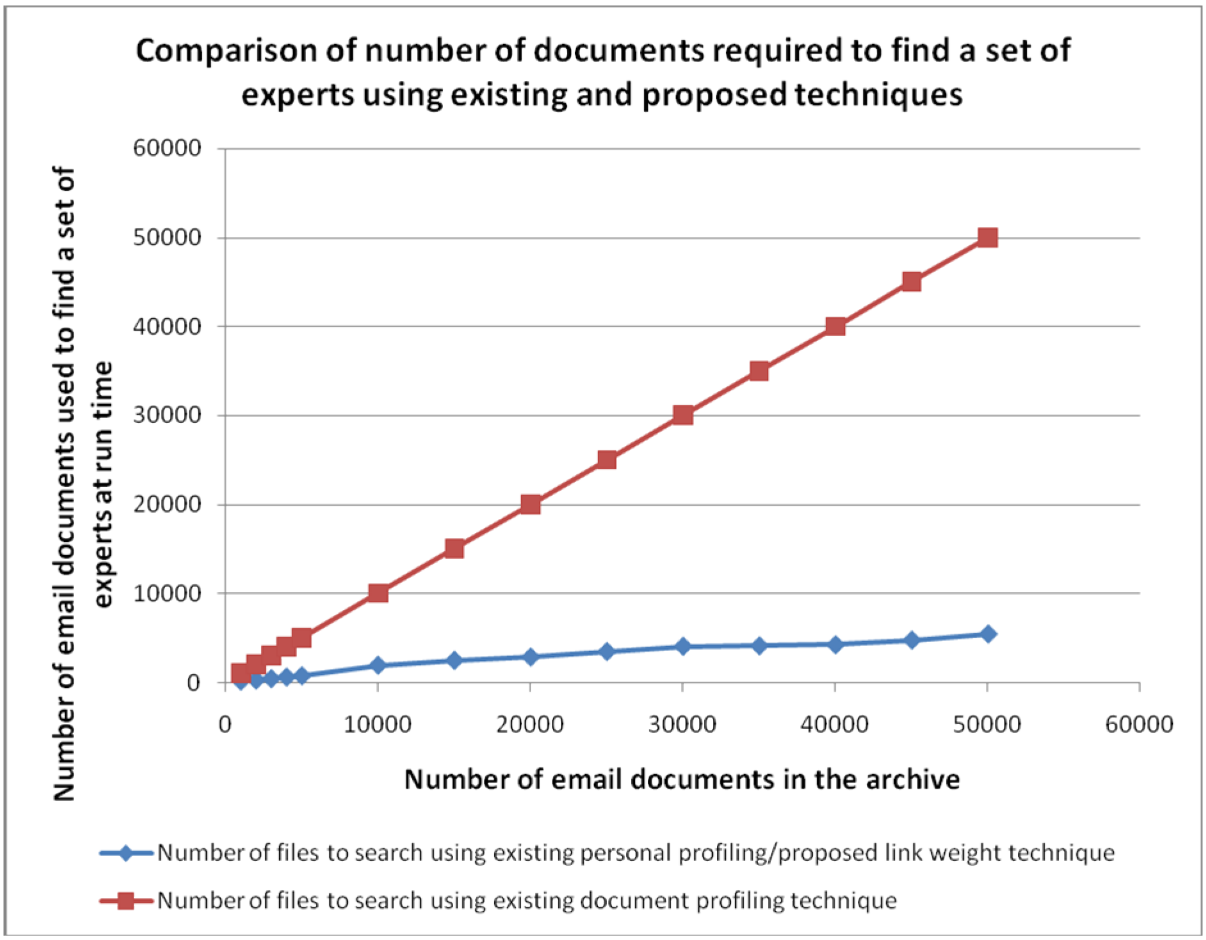


Figure 7: Comparison of number of documents required to find a set of experts using existing and proposed techniques

Table 2: Comparison of number of documents required to find a set of experts using existing and proposed techniques

NO.	Total number of files in archive	Number of files to be searched using existing document profiling technique	Number of files to be searched using existing personal profiling/ proposed link weight technique
1	1000	1000	**163**
2	2000	2000	**284**
3	3000	3000	**443**
4	4000	4000	**620**
5	5000	5000	**761**
6	10000	10000	**1934**
7	15000	15000	**2481**
8	20000	20000	**2864**
9	25000	25000	**3469**
10	30000	30000	**4055**
11	35000	35000	**4107**
12	40000	40000	**4223**
13	45000	45000	**4749**
14	50000	50000	**5432**

Fig. 7 and Table 2 gives an idea of the relationship between the number of documents to be searched in the existing document profiling technique compared to existing personal profiling/proposed link weight techniques. In Fig. 7, the total number of files in the archive is given by the X-axis and the total number of files that are processed to find an expert using respective techniques is shown on Y-axis. Fig 7 also shows that, as the total number of the emails in archive increases, the number of files that are required to be searched for finding an expert using existing personal profiling or proposed link weight techniques increases with a lower constant for the slope.

Table 3: Comparison of time required to find a set of experts using existing and proposed techniques

Time to find an expert / No. of documents	Existing document profiling technique (DP)	Existing personal profiling technique (PP)	Proposed link weight technique (LW)	% of time increase from PP to LW i.e. ((LW-PP) / PP)*100	% of time increase from PP to DP i.e. ((DP-PP) / PP)*100
1000	2	0.96	1	4.16	100
2000	3	1	1.67	67	79.64
3000	4	1.39	2.06	48.20	94.17
4000	4.72	1.86	2.33	25.26	102.57
5000	5.92	2.13	3	40.84	97.33
10000	11.99	5.8	6.26	7.93	91.53
15000	18.87	7.8	8.46	8.46	123.04
20000	24.47	10.06	10.73	6.66	128.05
25000	30.87	13.46	14.13	4.97	118.47
30000	37.8	16.53	17.00	2.84	122.35
35000	43.53	17.6	18.33	4.14	137.47
40000	51.53	19	19.93	4.89	158.55
45000	60.51	20.53	21.40	4.23	182.75
50000	68.47	22.93	23.66	3.18	189.39

[shaded box] shows that as the archive size inscreases the percentage inscrease in the time for proposed link weight approach decreases (second last column) and the percentage inscrease in the time for existing document profiling approach inscreases (last column)

The numbers of files that are processed to find the expert are the same in the existing personal profile approach and the proposed link weight approach. However the proposed algorithm does the extra step of re ranking the experts. Therefore if the time taken by the proposed link weight approach is close to the existing personal profiling technique than that should be considered as satisfactory results in terms of time.

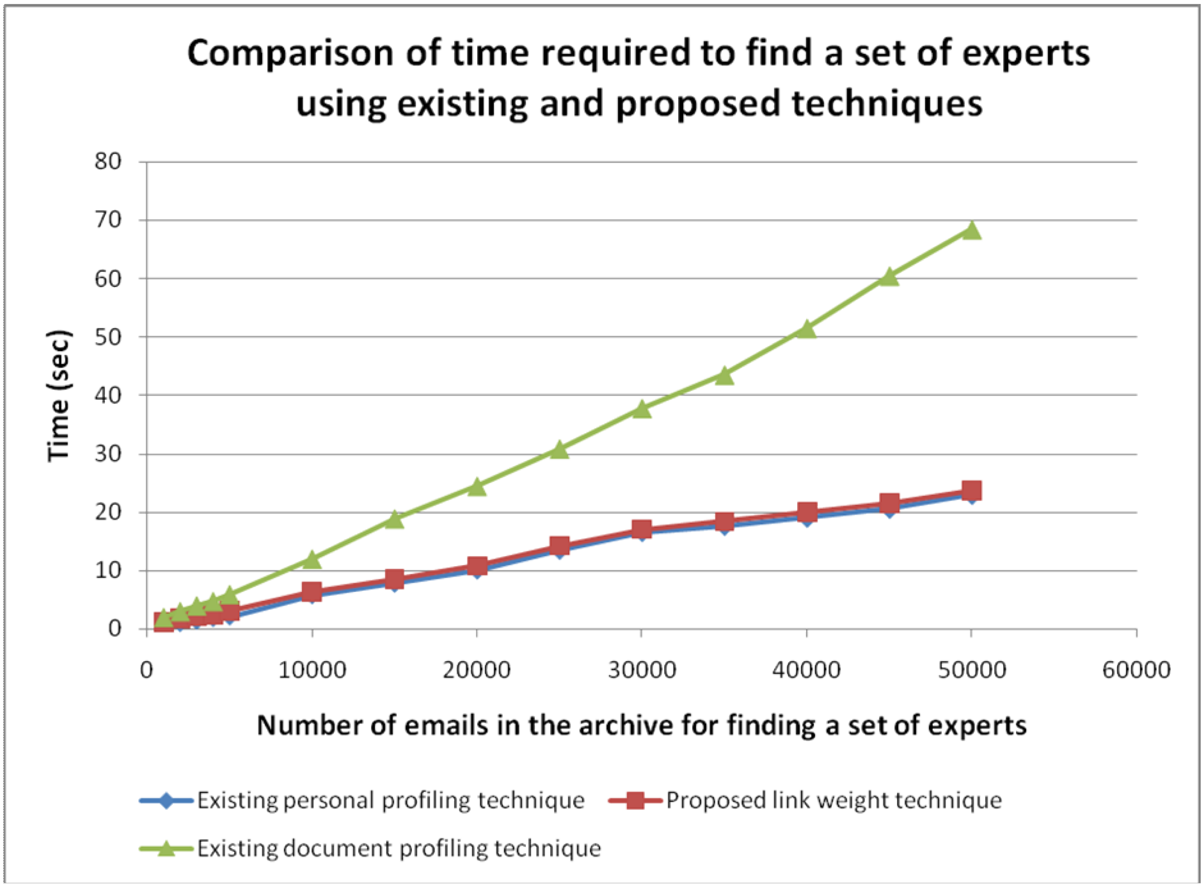


Figure 8: Comparison of time required to find a set of experts using existing and proposed techniques

Table 3 and Fig 8 give us the comparison of the time required by different searching techniques to find an expert from the given archive. The comparison is made for different size of the archive shown on the X-axis. From the last two columns of Table 3, it can be inferred that as the number of files in the archive increase, the percentage of increase in the time to find a set of experts between existing personal profiling technique and proposed link weight technique reduces. For the archive size of 50000 documents, the percentage of increase in the time to find a set of experts between existing personal profiling technique and proposed link weight technique is 3.18%. Also, from the last two columns of Table 3 it can be inferred that as the number of files in the archive increase, the percentage of increase in the time to find a set of experts between existing document profiling technique and proposed link weight technique increases drastically. For the archive size of 50000 documents, the percentage of increase in the time to find a set of experts between existing document profiling technique and proposed link weight technique is 189.39%. It can be seen that the difference in the time taken by the existing profiling method and proposed link weight approach is negligible. Since, the proposed algorithm takes the results of the existing personal profiling technique as input and gives the ranked list of experts as output, it strengthens the fact that the new link weight approach has minimal time penalty when used with the existing searching techniques. Thereby, the results of existing searching algorithm can be improved at the cost of minimal time penalty.

5.2 Analysis of results

The quality of the results from the proposed link weight technique was compared against the existing personal profiling technique. Three different runs were made using different keyword search phrases. Following is the analysis of all the three runs.

Search phrase: **computer network**
Size of the archive: 50000+
Weights:
Receiver: 1
Cc: 0.5
Sender: 0.1

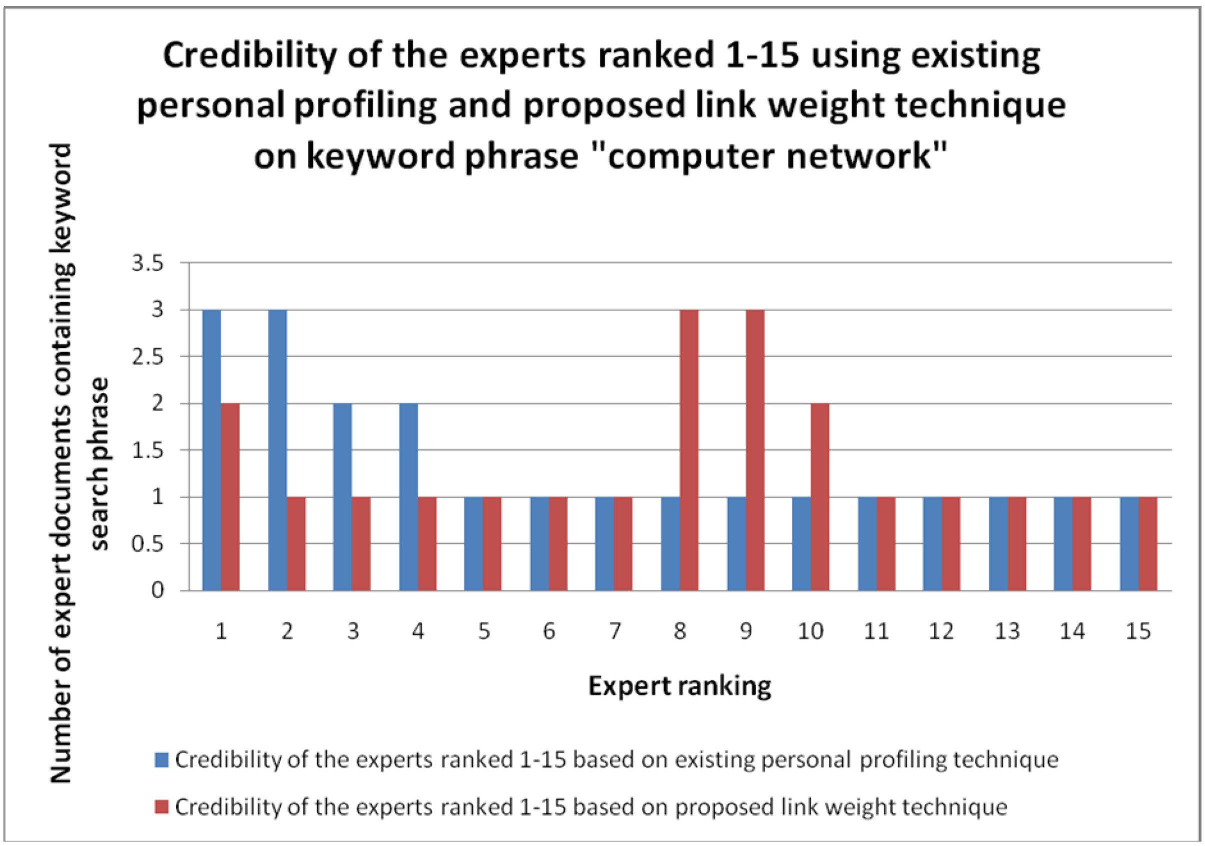


Figure 9: Credibility of the experts ranked 1-15 using existing personal profiling and proposed link weight technique on keyword phrase "computer network"

Fig 9 shows the experts ranked from 1-15 and the number of emails that contain the keyword search phrase for each individual expert. The number of emails containing the search phrase is known as the credibility of the expert. Credibility of the experts ranked from 1-15 using existing personal profiling technique and the proposed link weight technique is shown.

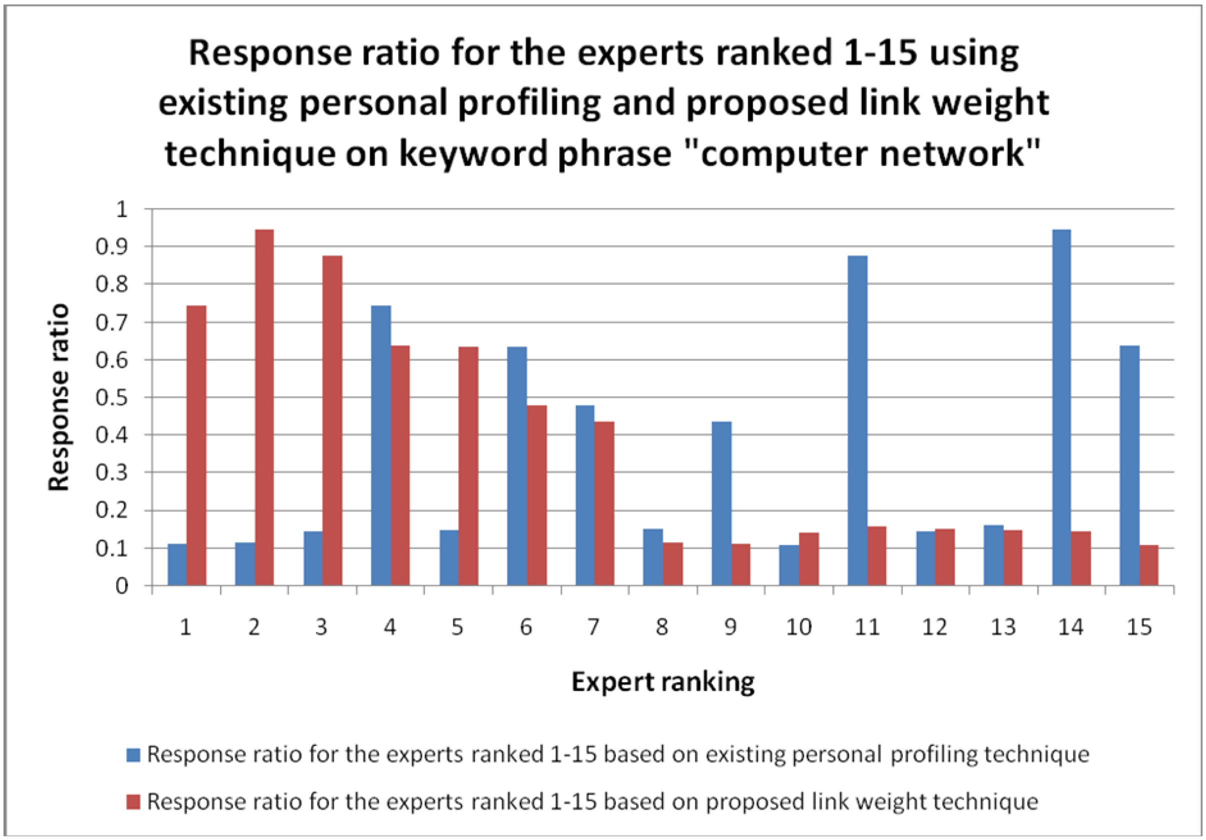


Figure 10: Response ratio for the experts ranked 1-15 using existing personal profiling and proposed link weight technique on keyword phrase "computer network"

The response ratio is an important parameter indicating the responsive behavior of the expert under consideration. Fig 10 displays the response ratio of the experts ranked 1-15 using the existing personal profile based technique and the proposed link weight approach. It can be seen that for the top five experts, the response ratio of the experts ranked by the link weight approach (red) is higher compared to the response ratio of the experts ranked by the existing personal profiling technique (blue). Thus, the list of experts given by the proposed link weight approach is more likely to respond to the user query than the list of experts found using the existing personal profiling approach. Fig 9 and Fig 10 are correlated; rank 1 using existing personal profile technique in Fig 9 is the same expert that corresponds to the rank 1 using existing personal profile technique in Fig 10. Similarly, rank 1 using proposed link weight technique in Fig 9 is the same expert that corresponds to the rank 1 using proposed link weight technique in Fig 10.

The experts ranked from 1-15 in figures 9 and 10 are same set of experts. In each figure the experts are ranked using two different approaches. The experts ranked 1 by both the approach may not be the same person.

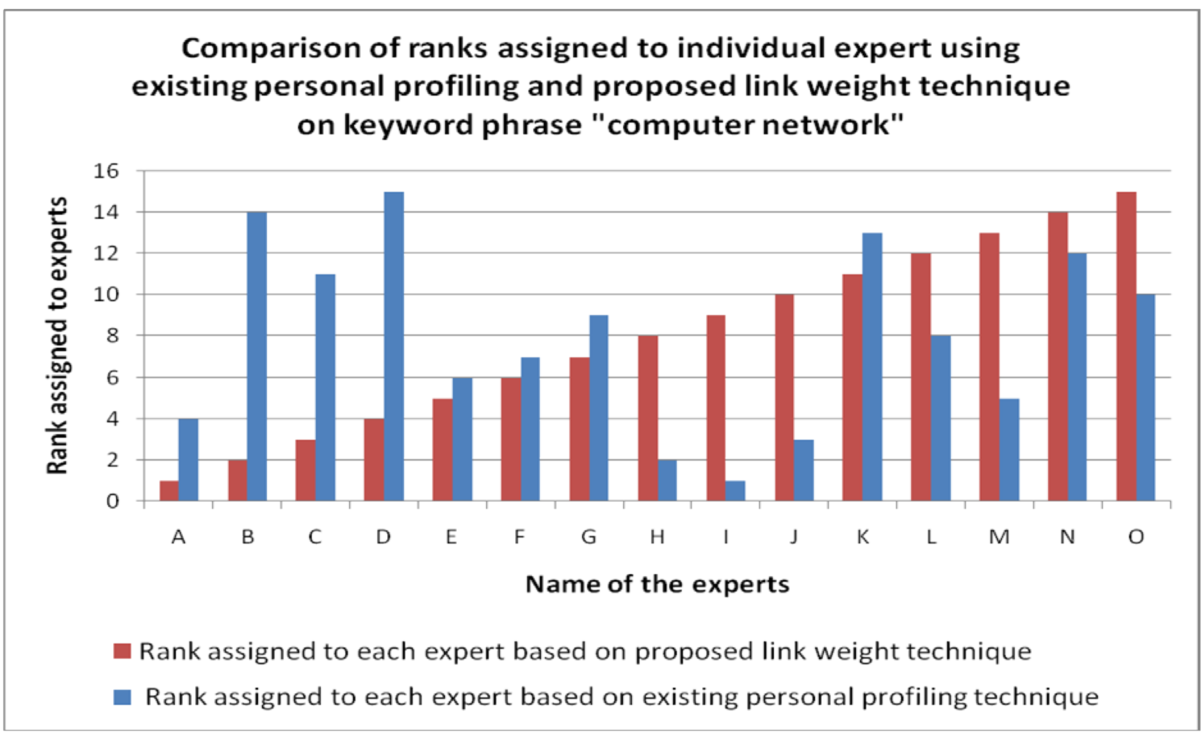


Figure 11: Comparison of ranks assigned to individual expert using existing personal profiling and proposed link weight technique on keyword phrase "computer network"

The ranking of experts in the proposed link weight approach is essentially a rearrangement of the ranking of experts in the existing personal profiling approach. Fig 11 gives us the rank of an individual expert for a given keyword phrase using the proposed link weight approach and compares it with the rank that was assigned to the same expert using the existing personal profiling technique. i.e. expert A was assigned rank 1 using proposed link weight technique while the same expert A was assigned rank 4 using existing personal profiling technique.

In Fig 9, it can be seen that the expert with “rank 1” using existing personal profiling approach has the highest credibility. Fig 10 shows that the same expert has a very low response ratio. For this reason, it can be seen that in Fig 11 the same expert is ranked 9 using the proposed link weight approach. This shows that an expert with lower response ratio is pushed to the lower ranks. Thus, it can be concluded that the new link weight based approach provides the user with more responsive experts.

Table 4: Detail analysis of the keyword search phrase “computer network”

Name of the expert	Rank based on proposed link weight technique (LW)	Rank based on existing personal profiling technique (PP)	Delta rank (LW - PP)	Response ratio
A	1	4	-3	0.745
B	**2**	**14**	**-12**	**0.947**
C	**3**	**11**	**-8**	**0.877**
D	**4**	**15**	**-11**	**0.639**
E	5	6	-1	0.637
F	6	7	-1	0.48
G	7	9	-2	0.438
H	**8**	**2**	**6**	**0.117**
I	**9**	**1**	**8**	**0.112**
J	**10**	**3**	**7**	**0.144**
K	11	13	-2	0.16
L	12	8	4	0.153
M	**13**	**5**	**8**	**0.15**
N	14	12	2	0.145
O	15	10	5	0.109

(Light grey) Represents the experts with a high response ratio and hence are ranked up in the list using the proposed link weight approach

(Dark grey) Represents the experts with a low response ratio and hence are pushed down the list using the existing personal profiling technique

Table 4 represents detailed analysis for the keyword search phrase “computer network” using the proposed link weight technique for ranking; the experts with a higher response ratio are given a better ranking compared to the experts with a lower response ratio. It can be seen that “expert B” is assigned rank 14 using the existing personal profiling technique while the same “expert B” is assigned rank 2 using the proposed link weight technique. The delta in the rank for “expert B” using two different techniques is negative and has a high value because the response ratio for “expert B” is 0.947 which is very high compared to many experts in the list. Hence, the “expert B” moves up in the list by 12 ranks. Similar is the case with “expert C” and “expert D”. Also, “expert M” is assigned rank 5 using the existing personal profiling technique while the same “expert M” is assigned rank 13 using the proposed link weight technique. The delta in the rank for “expert M” using two different techniques is positive and has a high value because the response ratio for “expert M” is 0.15 which is very low compared to many experts in the list. Hence, the “expert M” is pushed down in the list by 8 ranks. Similar is the case with “expert H”, “expert I” and “expert J”.

Search phrase: **operating system**
Size of the archive: 50000+
Weights:

Receiver:	1
Cc:	0.5
Sender:	0.1

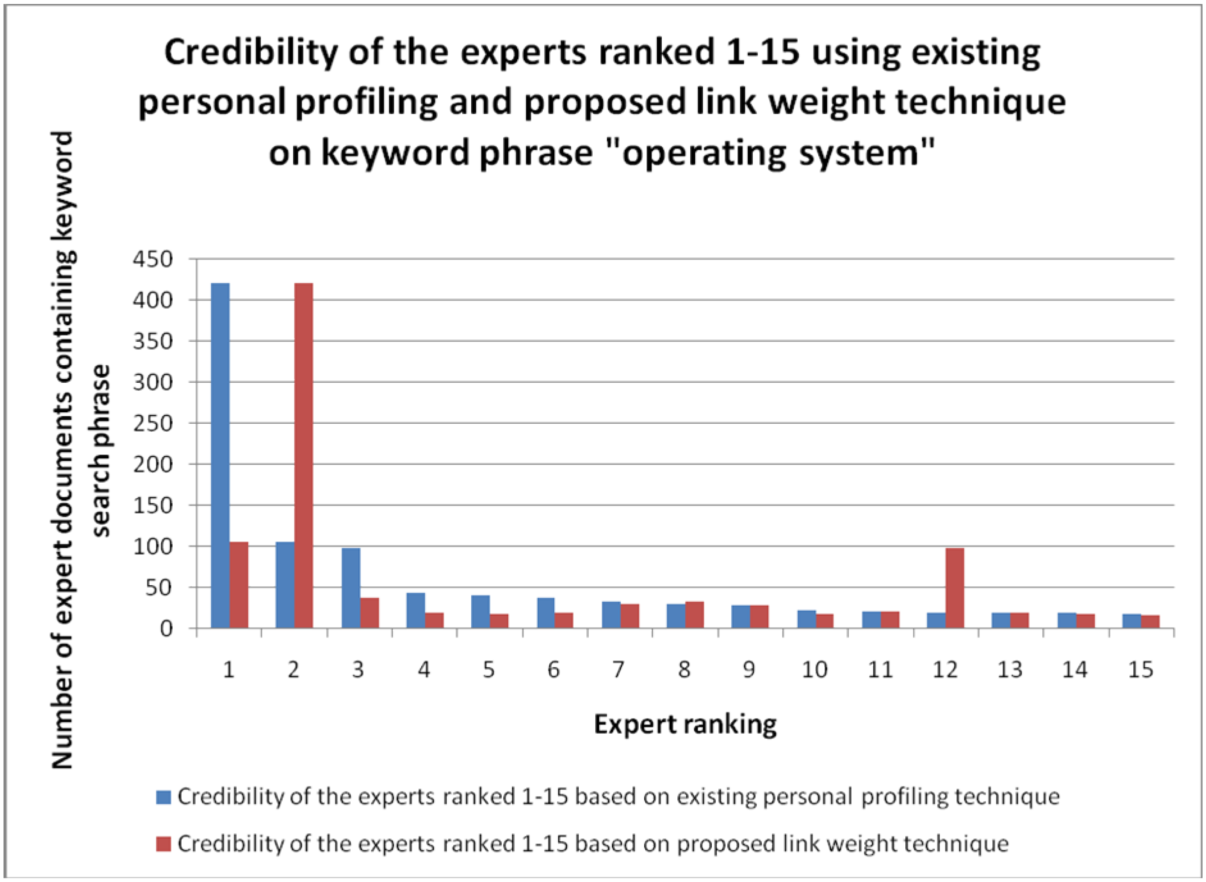


Figure 12: Credibility of the experts ranked 1-15 using existing personal profiling and proposed link weight technique on keyword phrase "operating system"

Fig 12 shows the experts ranked from 1-15 and the number of emails that contain the keyword search phrase for each individual expert. The number of emails containing the search phrase is known as the credibility of the expert. Credibility of the experts ranked from 1-15 using existing personal profiling technique and the proposed link weight technique is shown.

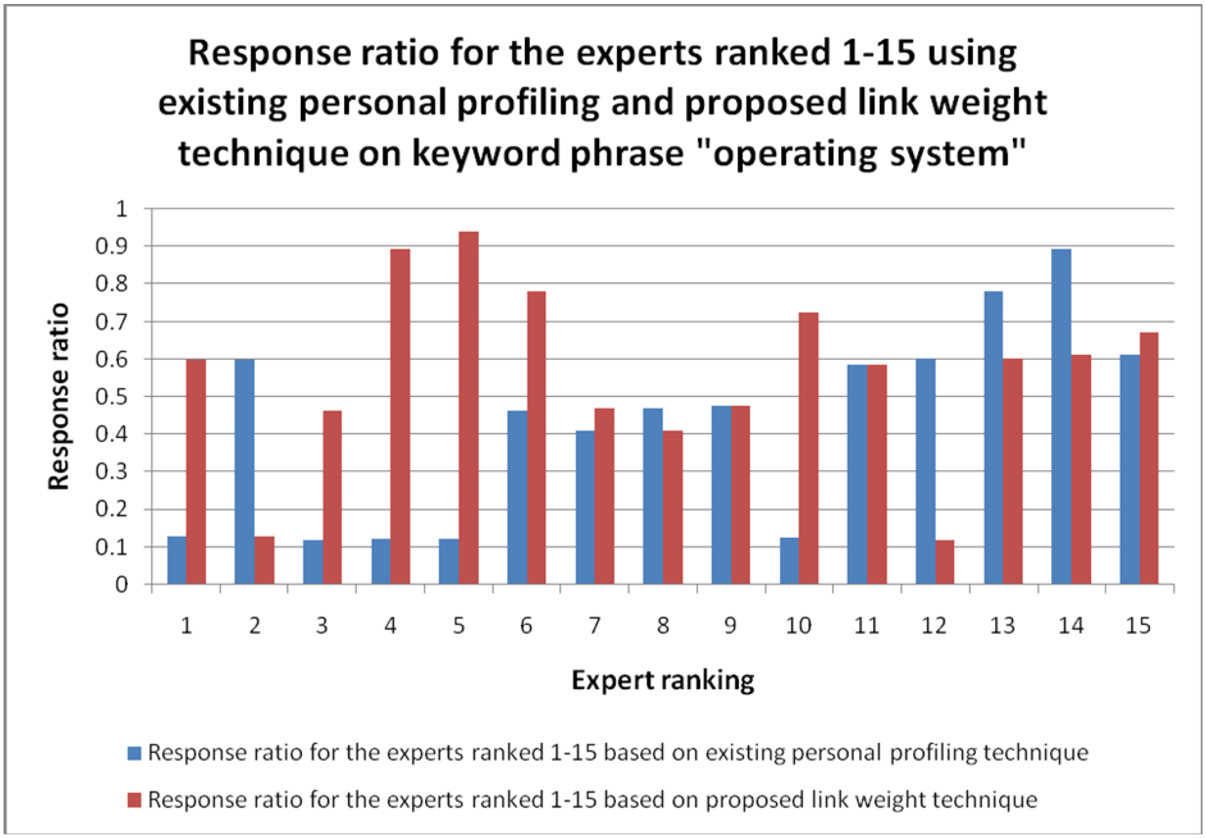


Figure 13: Response ratio for the experts ranked 1-15 using existing personal profiling and proposed link weight technique on keyword phrase "operating system"

The response ratio is an important parameter indicating the responsive behavior of the expert under consideration. Fig 13 displays the response ratio of the experts ranked 1-15 using the existing personal profile based technique and the proposed link weight approach. It can be seen that for the top five experts, the response ratio of the experts ranked by the link weight approach (red) is higher compared to the response ratio of the experts ranked by the existing personal profiling technique (blue). Thus, the list of experts given by the proposed link weight approach is more likely to respond to the user query than the list of experts found using the existing personal profiling approach. Fig 12 and Fig 13 are correlated; rank 1 using existing personal profile technique in Fig 12 is the same expert that corresponds to the rank 1 using existing personal profile technique in Fig 13. Similarly, rank 1 using proposed link weight technique in Fig 12 is the same expert that corresponds to the rank 1 using proposed link weight technique in Fig 13.

The experts ranked from 1-15 in figures 12 and 13 are same set of experts. In each figure the experts are ranked using two different approaches. The experts ranked 1 by both the approach may not be the same person.

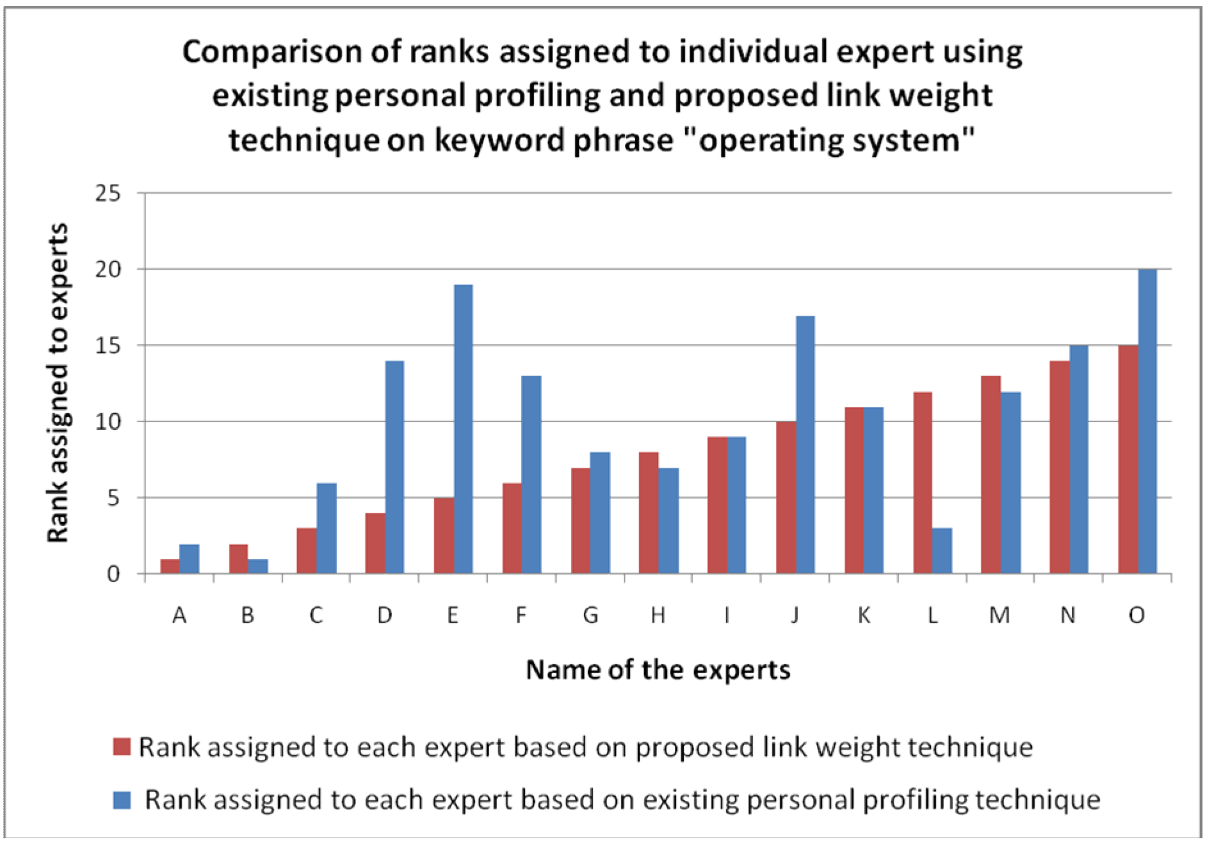


Figure 14: Comparison of ranks assigned to individual expert using existing personal profiling and proposed link weight technique on keyword phrase "operating system"

The ranking of experts in the proposed link weight approach is essentially a rearrangement of the ranking of experts in the existing personal profiling approach. Fig 14 gives us the rank of an individual expert for a given keyword phrase using the proposed link weight approach and compares it with the rank that was assigned to the same expert using the existing personal profiling technique. i.e. expert D was assigned rank 4 using proposed link weight technique while the same expert D was assigned rank 14 using existing personal profiling technique.

In Fig 12, it can be seen that the expert with “rank 3” using existing personal profiling approach has the highest credibility. Fig 13 shows that the same expert has a very low response ratio. For this reason, it can be seen that in Fig 14 the same expert is ranked 12 using the proposed link weight approach. This shows that an expert with lower response ratio is pushed to the lower ranks. Thus, it can be concluded that the new link weight based approach provides the user with more responsive experts.

Table 5: Detail analysis of the keyword search phrase "operating system"

Name of the expert	Rank based on proposed link weight technique (LW)	Rank based on existing personal profiling technique (PP)	Delta rank (LW - PP)	Response ratio
A	1	2	-1	0.599
B	2	1	1	0.128
C	3	6	-3	0.465
D	**4**	**14**	**-10**	**0.893**
E	**5**	**19**	**-14**	**0.942**
F	**6**	**13**	**-7**	**0.781**
G	7	8	-1	0.471
H	8	7	1	0.409
I	9	9	0	0.476
J	10	17	-7	0.726
K	11	11	0	0.586
L	**12**	**3**	**9**	**0.118**
M	13	12	1	0.602
N	14	15	-1	0.612
O	15	20	-5	0.672

(light grey)	Represents the experts with a high response ratio and hence are ranked up in the list using the proposed link weight approach
(dark grey)	Represents the experts with a low response ratio and hence are pushed down the list using the existing personal profiling technique

Table 5 represents detailed analysis for the keyword search phrase "operating system" using the proposed link weight technique for ranking; the experts with a higher response ratio are given a better ranking compared to the experts with a lower response ratio. It can be seen that "expert E" is assigned rank 19 using the existing personal profiling technique while the same "expert E" is assigned rank 5 using the proposed link weight technique. The delta in the rank for "expert E" using two different techniques is negative and has a high value because the response ratio for "expert E" is 0.942 which is very high compared to many experts in the list. Hence, the "expert E" moves up in the list by 14 ranks. Similar is the case with "expert D" and "expert F". Also "expert L" is assigned rank 3 using the existing personal profiling technique while the same “expert L” is assigned rank 12 using the proposed link weight technique. The delta in the rank for “expert L” using two different techniques is positive and has a high value because the response ratio for “expert L” is 0.118 which is very low compared to many experts in the list. Hence, the “expert L” is pushed down in the list by 9 ranks.

Search phrase: **database system**
Size of the archive: 50000+
Weights:
Receiver: 1
Cc: 0.5
Sender: 0.1

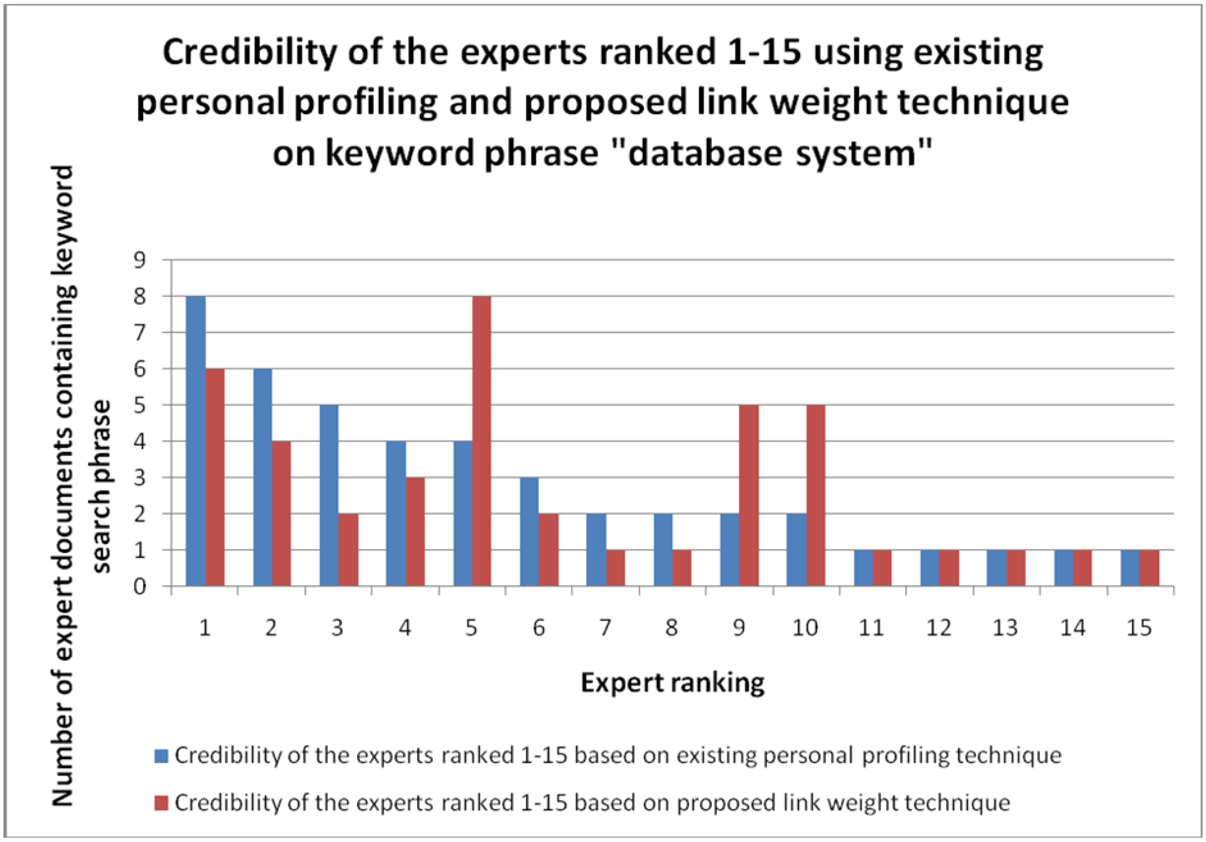


Figure 15: Credibility of the experts ranked 1-15 using existing personal profiling and proposed link weight technique on keyword phrase "database system"

Fig 15 shows the experts ranked from 1-15 and the number of emails that contain the keyword search phrase for each individual expert. The number of emails containing the search phrase is known as the credibility of the expert. Credibility of the experts ranked from 1-15 using existing personal profiling technique and the proposed link weight technique is shown.

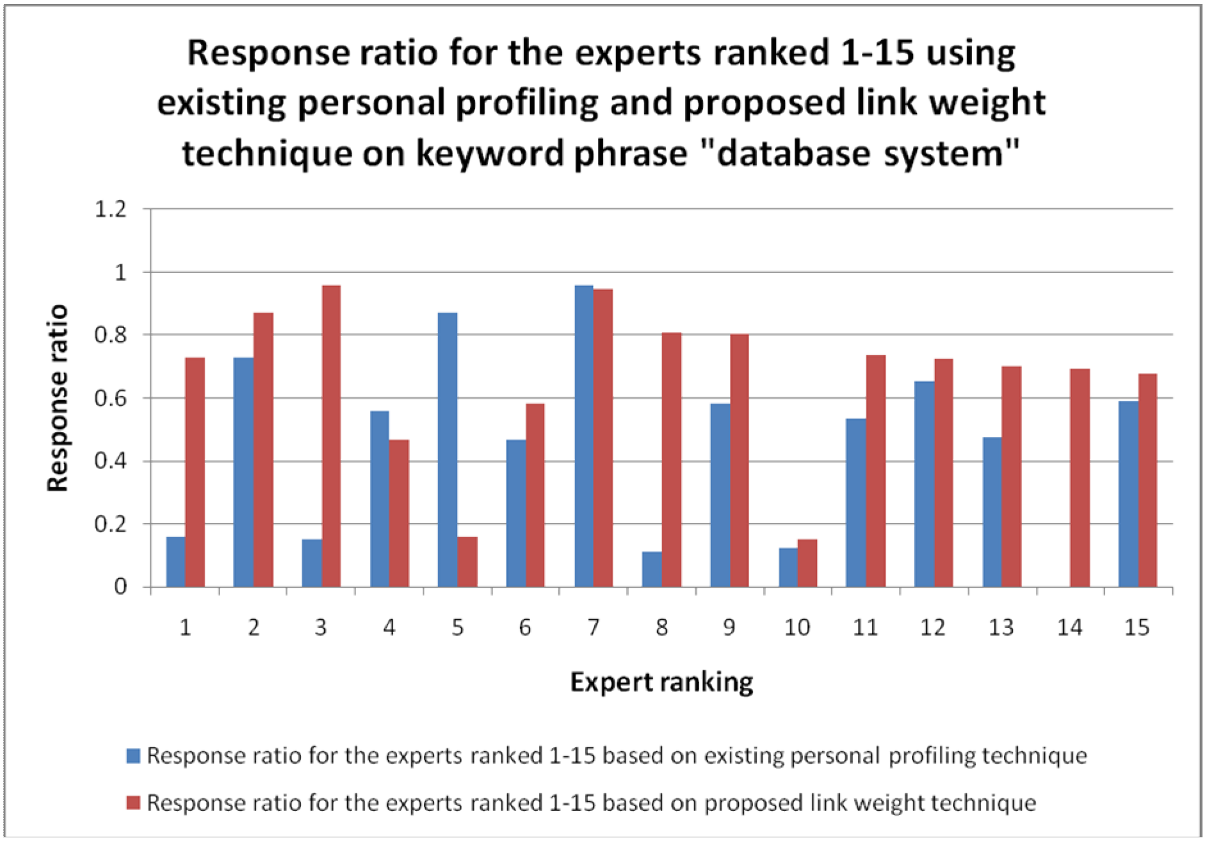


Figure 16: Response ratio for the experts ranked 1-15 using existing personal profiling and proposed link weight technique on keyword phrase "database system"

The response ratio is an important parameter indicating the responsive behavior of the expert under consideration. Fig 16 displays the response ratio of the experts ranked 1-15 using the existing personal profile based technique and the proposed link weight approach. It can be seen that for the top five experts, the response ratio of the experts ranked by the link weight approach (red) is higher compared to the response ratio of the experts ranked by the existing personal profiling technique (blue). Thus, the list of experts given by the proposed link weight approach is more likely to respond to the user query than the list of experts found using the existing personal profiling approach. Fig 15 and Fig 16 are correlated; rank 1 using existing personal profile technique in Fig 15 is the same expert that corresponds to the rank 1 using existing personal profile technique in Fig 16. Similarly, rank 1 using proposed link weight technique in Fig 15 is the same expert that corresponds to the rank 1 using proposed link weight technique in Fig 16.

The experts ranked from 1-15 in figures 15 and 16 are same set of experts. In each figure the experts are ranked using two different approaches. The experts ranked 1 by both the approach may not be the same person.

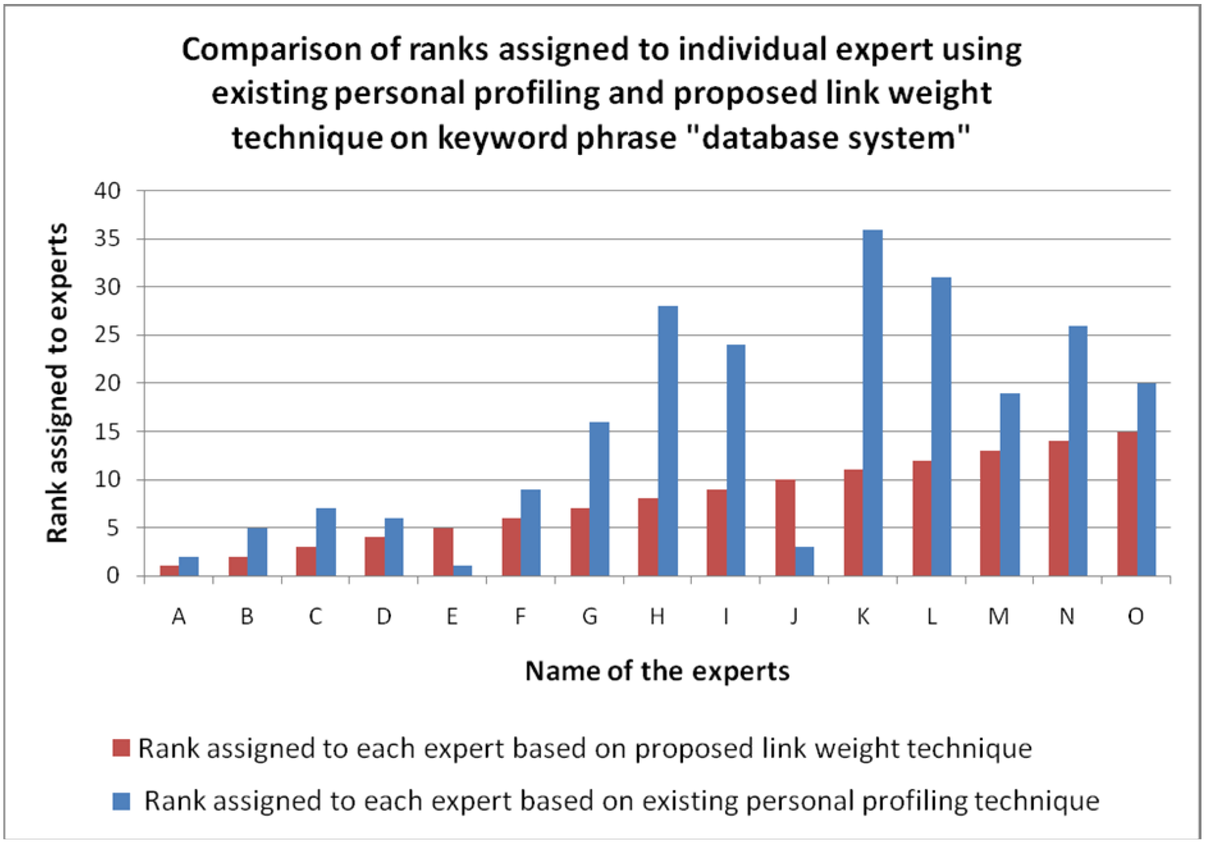


Figure 17: Comparison of ranks assigned to individual expert using existing personal profiling and proposed link weight technique on keyword phrase "database system"

The ranking of experts in the proposed link weight approach is essentially a rearrangement of the ranking of experts in the existing personal profiling approach. Fig 17 gives us the rank of an individual expert for a given keyword phrase using the proposed link weight approach and compares it with the rank that was assigned to the same expert using the existing personal profiling technique. i.e. expert A was assigned rank 1 using proposed link weight technique while the same expert A was assigned rank 2 using existing personal profiling technique.

In Fig 15, it can be seen that the expert with “rank 3” using existing personal profiling approach has the highest credibility. Fig 16 shows the same expert has a very low response ratio. For this reason, it can be seen that in Fig 17 the same expert is ranked 10 using the proposed link weight approach. This shows that an expert with lower response ratio is pushed to the lower ranks. Thus, it can be concluded that the new link weight based approach provides the user with more responsive experts.

Table 6: Detail analysis of the keyword search phrase "database system"

Name of the expert	**Rank based on proposed link weight technique (LW)**	**Rank based on existing personal profiling technique (PP)**	**Delta rank (LW - PP)**	**Response ratio**
A	1	2	-1	0.73
B	2	5	-3	0.872
C	3	7	-4	0.96
D	4	6	-2	0.466
E	5	1	4	0.159
F	6	9	-3	0.583
G	7	16	-9	0.948
H	**8**	**28**	**-20**	**0.808**
I	**9**	**24**	**-15**	**0.802**
J	10	3	7	0.152
K	**11**	**36**	**-25**	**0.738**
L	**12**	**31**	**-19**	**0.726**
M	13	19	-6	0.702
N	14	26	-12	0.695
O	15	20	-5	0.678

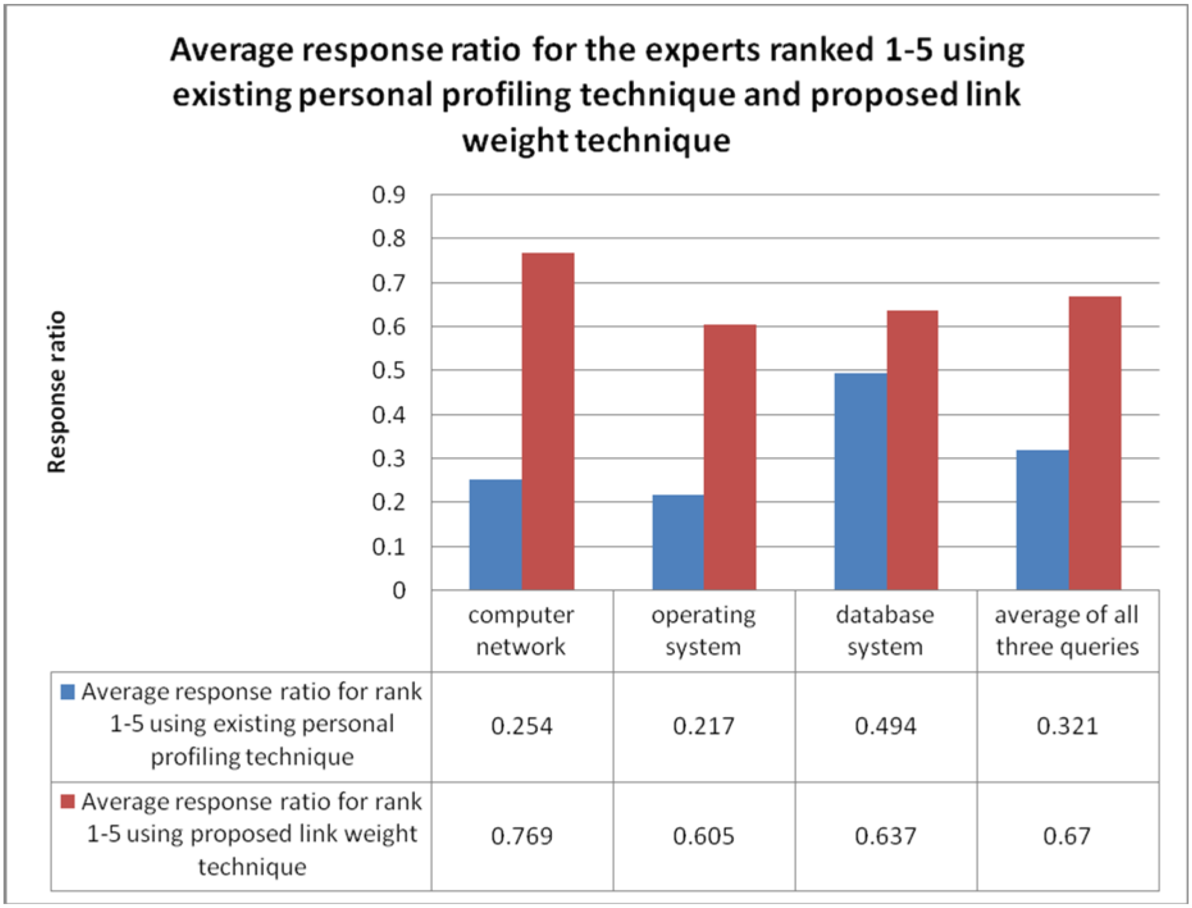
(Light grey) Represents the experts with a high response ratio and hence are ranked up in the list using the proposed link weight approach

(Dark grey) Represents the experts with a low response ratio and hence are pushed down the list using the existing personal profiling technique

Table 6 represents detailed analysis for the keyword search phrase "database system" using the proposed link weight technique for ranking; the experts with a higher response ratio are given a better ranking compared to the experts with a lower response ratio. It can be seen that "expert H" is assigned rank 28 using the existing personal profiling technique while the same "expert H" is assigned rank 8 using the proposed link weight technique. The delta in the rank for "expert H" using two different techniques is negative and has a high value because the response ratio for "expert H" is 0.808 which is very high compared to many experts in the list. Hence, the "expert H" moves up in the list by 20 ranks. Similar is the case with "expert I", "expert K" and "expert L". Also "expert J" is assigned rank 3 using the existing personal profiling technique while the same "expert J" is assigned rank 10 using the proposed link weight technique. The delta in the rank for "expert J" using two different techniques is positive and has a high value because the response ratio for "expert J" is 0.152 which is very low compared to many experts in the list. Hence, the "expert J" is pushed down in the list by 7 ranks. Similar is the case with "expert E".

Table 7: Average response ratio for the experts ranked 1-5 using existing personal profiling technique and proposed link weight technique

Response ratio / **keyword search phrase**	**Existing personal profiling technique (PP)**	**Proposed link weight technique (LW)**
computer network	0.254	0.769
operating system	0.217	0.605
database system	0.494	0.637
Average of all three queries	0.321	0.67



	computer network	operating system	database system	average of all three queries
Average response ratio for rank 1-5 using existing personal profiling technique	0.254	0.217	0.494	0.321
Average response ratio for rank 1-5 using proposed link weight technique	0.769	0.605	0.637	0.67

Figure 18: Average response ratio for the experts ranked 1-5 using existing personal profiling technique and proposed link weight technique

From Table 7 and Fig. 18 it can be seen that for all the three keyword search phrases, the average response ratio for the experts ranked 1-5 using the proposed link weight technique is always greater than the response ratio for the experts ranked 1-5 using the existing personal profiling technique. For query “computer network” the average response ratio for the experts ranked 1-5 using existing personal profiling technique is 0.254 while the average response ratio for the experts ranked 1-5 using proposed link weight technique is 0.769. The top five experts using the existing personal profiling technique are expected to respond 24.4% of the time while the top five experts using the proposed link weight technique are expected to respond 76.9% of the time. Thus, for “computer network” the top five experts ranked using proposed link weight technique are expected to reply 52.5% times more than the top five experts ranked using the existing personal profiling technique. Even for the average of all the three queries, the top five experts ranked using proposed link weight technique are expected to reply 34.9% more than the top five experts ranked using the existing personal profiling technique.

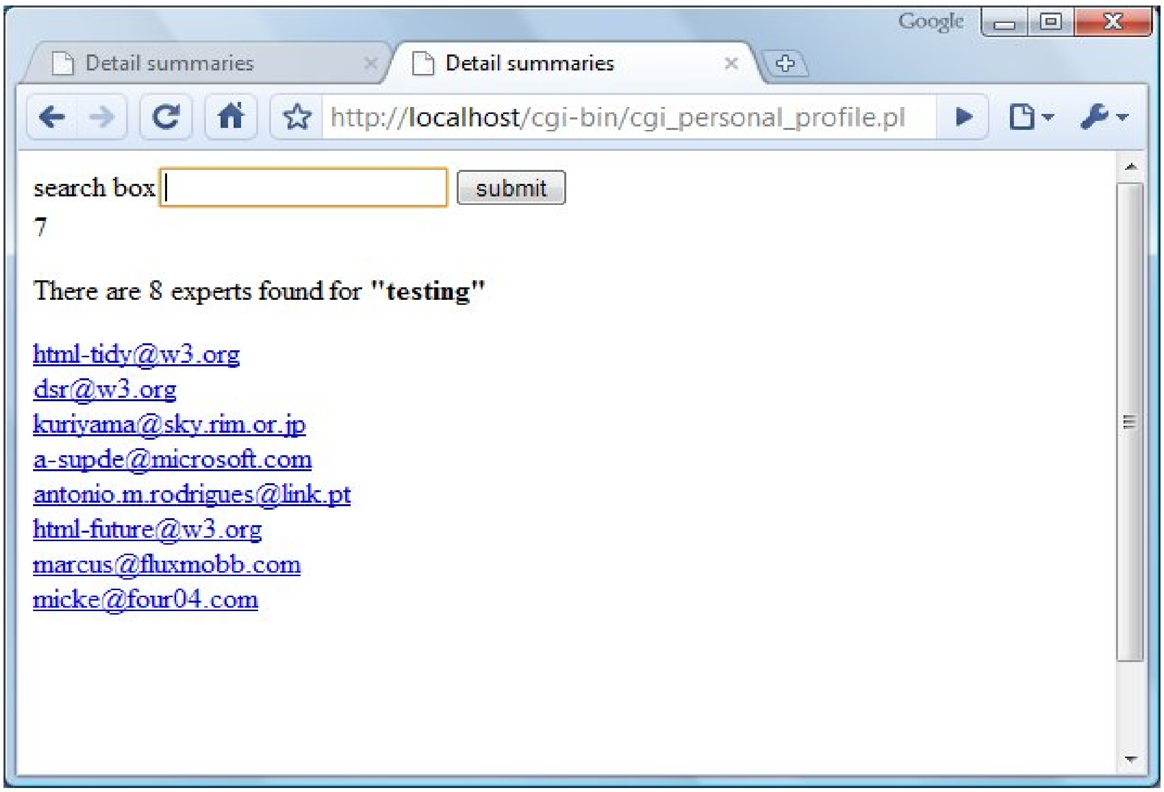


Figure 19: Final screen used for link weight based Expert finding system

The final screen used to give the input keyword phrase for the proposed link weight approach to expert searching is shown in Fig. 18. The screen consists of a search box and a submit button. Once the keyword is entered in the search box and the submit button is clicked, the link weight based expert search is performed and we get a list of experts that are arranged taking the response ratio into consideration.

6. CONCLUSION

In the study of the related systems in section 3, it is found that either the system has a good precision or it is more efficient. If we go on increasing the precision, the efficiency is reduced and the system is rendered unusable. Such systems are only used for theoretical purposes or for the purpose of comparing some new systems against it. If we increase the efficiency of the system, we obtain quicker results but the accuracy is compromised. If the degree of inaccuracy is high then the results used would have unfavorable consequences.

Table 7 and Fig. 18 shows that, in all the circumstances the response ratio of the top five experts found using the proposed link weight algorithm is at least 14.3% higher than the response ratio of the top five experts found using existing personal profiling technique. In some cases such as for the query “computer network”, the increase in the response ratio of the proposed link weight technique can be as high as 51.5%. Thus, it can be concluded that the proposed algorithm produces a set of experts that are more responsive to the user query compared to the existing techniques.

Also, the proposed link weight algorithm has a minimal time penalty when used with the existing expert finding techniques. The time efficiency of the proposed algorithm is comparable with the existing personal profiling technique for expert finding. From Table 3 and Fig 8, it can be seen that for an archive of 25000 email documents the percentage increase in the time to find a set of experts between existing personal profiling technique and proposed link weight technique is 4.97% and this value goes on decreasing as the size of the archive increases. According to table 3, the percentage increase in the time to find a set of experts between existing personal profiling technique and proposed link weight technique for the archive of 50000 email archive is 3.18%. This shows that as the size of the archive increases, the time difference between the proposed link weight algorithm and the existing personal profiling technique is negligible. Thus, finding an expert using proposed link weight technique provides the results with improved quality as compared to existing personal profiling technique with a minimal time penalty.

7. FUTURE WORK

The documents are an important part of the repository where all the knowledge is stored. These documents can be in varied formats i.e. doc, pdf, rtf, etc. These documents in different formats can be used to assist in the process of expert finding. When all these documents are used along with the proposed algorithm, better results are expected than the one obtained by using only the emails. Also, these days the trend of writing white papers is on a rise. Adding all these additional knowledge resources to the existing expert search engine will definitely improve the results. But these improved results will come with an additional cost of processing. Further research is required to use these additional knowledge documents in the expert search engine and find the results within permissible time limitations.

Rather than just analyzing the email one word at a time, we can process the email and try to find out some meaning. This would help us obtain better and more accurate results. Even if just the nature of the email is known i.e. positive/negative still it would be very helpful and would lead to more accurate ranking mechanism. Also, if the time difference between the email and its response is accurately known then the quality of the results would definitely improve.

If the nature of the organization where the system is to be implemented is known in advance, it would be easier to remove the relative stop words thereby reducing the size of the files to be searched for the experts. Also, the words that are given more importance in the organization should be known for improving the results. Abbreviation, synonym and organization slang if known while creating the system, can be used to make the system more efficient.

Appendices

Appendix A. Source Code

```
#! /bin/perl
#	lw_profile.pl
#	This code would concatenate the various files from the user into a
#	single file for each user and also create the .profile file for each user.
#	This file is also used to create the communication matrix.
#	(c) 2009 Parin Shah
##########################################################################
$files_processed = 0;
$count = 0;
$file_counter = 0;
my %connection_hash;
$sender_weight = 0.1;
$cc_weight = 0.5;

##########################################################################
#	This module concatenates all the files from a specific user and
#	creates a single .out file for each user
#	input: the email archive with all the .txt files
#	output: concatenated files with .out extention
##########################################################################
sub create_out_files {
		my @out_files = glob("./out/*.out");
		unlink @out_files;
#		open(INPUT_FILES, "<out.asd") or die "cannot open input file of experts";;
#		@input_files = <INPUT_FILES>;
		@input_files = glob("./d/*.txt");
		foreach(@input_files) {
			$count = 0;
			++$file_counter;
			open (IN, "<$_") or next;
			print "\n$file_counter  <===>  $_";
			while($count <= 16){
			$line = lc(<IN>);
			$line =~ s/\@/__at__/g;
			$line =~ s/\./__dot__/g;
			$line =~ s/\-/__dash__/g;
			if($line	=~	m/^email="([a-zA-Z0-9\_]+__at__[a-zA-Z0-
9\_]+__dot__[a-zA-Z0-9\_]+)"/) {
				$sender = $1;
			} elsif($line =~ m/subject="(.*)"/) {
				$subject = $1;
			} elsif ($line =~ m/^to:\s*(.*)/) {
```

```
				@to_array = ();
				@temp = split(/ /, $1);
				for ($i=0; $i< scalar(@temp); $i++) {
					if($temp[$i]    =~    m/([a-zA-Z0-9\_]+__at__[a-zA-Z0-
9\_]+__dot__[a-zA-Z0-9\_]+)/) {
						push(@to_array, $1);
						$connection_hash{$sender}{$1}                    =
$sender_weight + $connection_hash{$sender}{$1};
						++$connection_hash{$1}{$sender};
					}
				}
			} elsif ($line =~ m/^cc:\s*(.*)/) {
				@cc_array = ();
				@temp = split(/ /, $1);
				for ($i=0; $i< scalar(@temp); $i++) {
					if($temp[$i]    =~    m/([a-zA-Z0-9\_]+__at__[a-zA-Z0-
9\_]+__dot__[a-zA-Z0-9\_]+)/) {
						push(@cc_array, $1);
						$connection_hash{$sender}{$1}                    =
$sender_weight + $connection_hash{$sender}{$1};
						$connection_hash{$1}{$sender}  =  $cc_weight
+ $connection_hash{$1}{$sender};
					}
				}
			}
			++$count;
			}

			undef $/;
			$line = <IN>;
			$line =~ s/\n/ /g;
			$/ = "\n";

#			open (OUT_SENDER, ">>./out/$sender.out") or next;
#			print OUT_SENDER $line;
#			print OUT_SENDER "\n";
#			close(OUT_SENDER);

			foreach(@to_array) {
				open (OUT_TO, ">>./out/$_.out") or next;
				print OUT_TO $line;
				print OUT_TO "\n";
				close(OUT_TO);
			}

			foreach(@cc_array) {
```

```
                    open (OUT_CC, ">>./out/$_.out") or next;
                    print OUT_CC $line;
                    print OUT_CC "\n";
                    close(OUT_CC);
                }
        }
        close(IN);
        close(INPUT_FILES);
}                                       # end function create_out_files()


#####################################################################
#       This module creates the .profile files
#       input: the .out files of each users
#       output: .profile files of each users
#####################################################################
sub create_profile {
        my @profile_files = glob("./out/*.profile");
        unlink @profile_files;
        print "Starting profiling phase";
        @out_files = glob("./out/*.out");
        print "\n total number of out files", scalar(@out_files);
        foreach(@out_files){
                $words_in_file = 0;
                ++$files_processed;
                print "\nFile being processed $_ : $files_processed";
                %word = ();
                open (IN, "<$_") or next;
                $profile = $_;
                $profile =~ s/([a-zA-Z0-9\_]+).out/$1.profile/;
                open (PROFILE_OUT, ">$profile") or next;
                @line = <IN>;
                chomp(@line);
                $total_number_of_files = scalar(@line);
                foreach(@line) {
                        if(length($_) == 0) {
                                --$total_number_of_files;
                                next;
                        }
                        $_ =~ s/\n/ /g;
                        $_ =~ s/\/|\\|\+|\$/ /g;
                        $_ =~ s/[!@#^%&*()={}:;<>,.?"|_-]//g;
                        $_ =~ s/\s+[0-9]*\s+/ /g;
                        $_ =~ s/\s+/ /g;
                        @array = ();
                        @array = split(/ /,lc($_));
```

```
			$words_in_file = $words_in_file + scalar(@array);

			foreach $k(keys %word) {
				$word{$k}[1] = 0;
			}

			foreach $w (@array) {
				if (!exists($stop_word{$w})) {
					if(!defined($word{$w}[0])) {
						$word{$w}[0] = 0;
						$word{$w}[1] = 0;
						$word{$w}[2] = 0;
						$word{$w}[3] = 0;
					}
					++$word{$w}[0];
					if (($word{$w}[1]) != 1) {
						$word{$w}[1] = 1;
						++$word{$w}[2];
					}

				}
			}
		}

		foreach $k(keys %word) {
			$word{$k}[3] = $total_number_of_files;
		}

		@key2 = sort {$word{$b}[0] <=> $word{$a}[0]} keys %word;
		foreach(@key2) {
			print   PROFILE_OUT   "$_   ",   "$word{$_}[0]   ",   "$word{$_}[1]
","$word{$_}[2] ","$word{$_}[3] ","\n";
		}
	}
	close(PROFILE_OUT);
	close(IN);
}			# end function create_profile()


######################################################################
#	This module writes the communication matrix to the file for
#	later use
#	input:
#	output: connection_array.pds
######################################################################
sub create_hash {
```

```
	@h = keys %connection_hash;
	open (HASH, ">connection_array.pds") or die "cannot create the hash file";
	foreach $k (@h) {
		foreach $l (@h) {
			if (exists $connection_hash{$k}{$l}) {
				print HASH "$k\t\t$l\t\t$connection_hash{$k}{$l}\n"
			}
		}
	}
	close(HASH);
}

create_out_files();
create_hash();
create_profile();
close(OUT);

#! /bin/perl
#	lw_person_profile.pl
#	This code is used to find expert using the person profile approach
#	(c) 2009 Parin Shah
###########################################################################

$count = 0;
$total = 0;
$flag = 0;
$freq = 0;
$no_of_files = 0;
%user;
@search_keywords;
#$search_query = "for your";
$search_query = join(" ", @ARGV);
$string = lc($search_query);
@search_keywords = split(/ /, $string);
@files_to_process = glob("./out/*.profile");

foreach(@files_to_process) {
	open(IN, "<$_") or next;
	$_ =~ m/\.\/out\/([a-zA-Z0-9\_]+)\.profile/;
	@lines = <IN>;
	$count = 0;
	$total = 0;
	$min_freq_words = 99999;
	$min_freq_file = 99999;
	for($i=0;$i<scalar(@search_keywords);$i++) {
```

```
		foreach $l (@lines) {
			if($l =~ m/^$search_keywords[$i] /) {
				@temp = split(/ /, $l);
				if ($temp[1] < $min_freq_words) {
					$min_freq_words = $temp[1];
				}
				++$count;
				last;
			}
		}
	}

	if ($count == scalar(@search_keywords)) {
		$user{$1}[0] = $min_freq_words;
	}
}

@unprocessed_user = keys %user;
open (OUT, ">../../../../Users/PDS/Desktop/rank_based_on_profile.pds") or die "cannot make rank_based_on_profile.pds";
foreach $f (@unprocessed_user) {
	$final = "./out/$f.out";
	open(FINAL_UESR_OUT, "<$final") or die "Cannot find connection file...";
	@out_lines = <FINAL_UESR_OUT>;
	$freq = 0;
	$no_of_files = 0;
	foreach $l (@out_lines) {
	$line = lc($l);
		if(@c = $line =~ m/$string/g) {
			$new_requirement{$f}[0] = $new_requirement{$f}[0] + scalar(@c);
			$new_requirement{$f}[1] = ++$no_of_files;
		}
	}
}

foreach (sort {$new_requirement{$b}[1] <=> $new_requirement{$a}[1]} keys %new_requirement) {
	print OUT "$_ $new_requirement{$_}[0] $new_requirement{$_}[1]\n";
}


close(OUT);

#! /bin/perl
#	lw_link_weight_profile.pl
#	This code is used to find expert using the link weight approach
```

```
#         (c) 2009 Parin Shah
#####################################################################
$count = 0;
$total = 0;
%user;
@search_keywords;
%connection_hash;
#$connection_input = "connection_array.pds";
open (OUT, ">../../../../Users/PDS/Desktop/rank_based_on_link_weight.pds") or die
"CAnnot open";
#$search_query = "for your";
$search_query = join(" ", @ARGV);
$string = lc($search_query);
@search_keywords = split(/ /, $string);
open(CONNECTION_FILE, "<connection_array.pds") or die "Cannot find connection
file...";
@temp = <CONNECTION_FILE>;
foreach(@temp) {
        @temp1 = split(/\t\t/ , $_);
        $connection_hash{$temp1[0]}{$temp1[1]} = $temp1[2];
}

@files_to_process = glob("./out/*.profile");

foreach(@files_to_process) {
        open(IN, "<$_");
        $_ =~ m/\.\/out\/([a-zA-Z0-9\_]+)\.profile/;
        @lines = <IN>;
        $count = 0;
        $total = 0;
        for($i=0;$i<scalar(@search_keywords);$i++) {

                foreach $l (@lines) {
                        if($l =~ m/^$search_keywords[$i] /) {
                                @temp = split(/ /, $l);
                                ++$count;
                                last;
                        }
                }
        }

        if ($count == scalar(@search_keywords)) {
                push (@{$user{$1}} , @temp);
        }
}
```

```
@unprocessed_user = keys %user;

foreach $first (@unprocessed_user) {
	foreach $second (@unprocessed_user) {
		$sent_receive{$first}[0]	=	$sent_receive{$first}[0]	+
$connection_hash{$first}{$second}; #sent
		$sent_receive{$first}[1]	=	$sent_receive{$first}[1]	+
$connection_hash{$second}{$first}; #receive
	}
	if ($sent_receive{$first}[1] == 0) {
		$sent_receive{$first}[1] = 1 + $sent_receive{$first}[1]
	}
	$sent_receive_ratio = $sent_receive{$first}[0]/($sent_receive{$first}[1]);
	if($sent_receive_ratio < 1) {
		$sent_receive{$first}[2] = $sent_receive_ratio;
		$sent_receive{$first}[3] = $sent_receive_ratio * $user{$first}[3];
	#multiplying with the total number of files containing the words
	} else {
		$sent_receive{$first}[2] = (1/$sent_receive_ratio);
		$sent_receive{$first}[3]	=	(1/$sent_receive_ratio)	*	$user{$first}[3];
	#multiplying with the total number of files containing the words
	}
}

@ratio_sorted_user	=	(sort	{$sent_receive{$b}[2]	<=>	$sent_receive{$a}[2]}	keys
%sent_receive);
foreach $f (@ratio_sorted_user) {
	$final = "./out/$f.out";
	open(FINAL_UESR_OUT, "<$final") or die "Cannot find connection file...";
	@out_lines = <FINAL_UESR_OUT>;
	$flag = 0;
	$freq = 0;
	$no_of_files = 0;
	foreach $l (@out_lines) {
		$line = lc($l);
		if(@c = $line =~ m/$string/g) {
			$new_requirement{$f}[0] = $new_requirement{$f}[0] + scalar(@c);
			$new_requirement{$f}[1] = ++$no_of_files;
		}
	}
	$new_requirement{$f}[2] = $sent_receive{$f}[2] * $new_requirement{$f}[1];
}

foreach	(sort	{$new_requirement{$b}[2]	<=>	$new_requirement{$a}[2]}	keys
%new_requirement) {
	if ($new_requirement{$_}[0] != NULL) {
```

```
		print	OUT	"$_	$new_requirement{$_}[0]	$new_requirement{$_}[1]
$new_requirement{$_}[2]		$sent_receive{$_}[0]		$sent_receive{$_}[1]
$sent_receive{$_}[2]\n";
	}
}

#!c:/Perl/bin/perl.exe
#Script created the GUI for the link weight approach
#for expert finding
#
#(C) 2009 Parin Shah. All Rights Reserved.
#Version 1.00		11/01/2009
############################################################################

############################################################################
#	Perl modules used in this program
############################################################################
use warnings;
use strict;
use CGI qw( :standard );

print header(), start_html( "Detail summaries" );
print '<form id="form1" name="form1" method="post" action="">';
print  '<label>search box';
print  '<input type="text" name="search_box" id="search_box" />';
print  '</label>';
print   '<input name="submit" type="submit" value="submit" />';
print '</form>';

#unlink "hello.txt";
my $var = param("search_box");
print length($var);
if (length($var) != 0) {
	system ("lw_link_weight_profile.pl $var");
	open (IN, "<../../../../Users/PDS/Desktop/rank_based_on_link_weight.pds") or die
"Cannot open the output file...";
	my @output = <IN>;
	my $no_of_expert = scalar(@output);
	if($no_of_expert > 15) {
		print "<p>There are 15 experts found for <b>\"",$var, "\"</b></p>";
	} else {
		print "<p>There are ",$no_of_expert,   " experts found for <b>\"",$var,
"\"</b></p>";
	}
	for(my $i = 0; $i <15; $i++) {
		my @temp = split(' ', $output[$i]);
```

```
		$temp[0] =~ s/__at__/\@/g;
		$temp[0] =~ s/__dot__/\./g;
		$temp[0] =~ s/__dash__/\-/g;
		print '<a href="mailto:', $temp[0], '">',$temp[0], '</a><br >';
	#	<a href="mailto:manish@simplygraphix.com">Send me an email</a>
	}
	close(IN);
}
print end_html();

#!c:/Perl/bin/perl.exe
#Script created the GUI for the personal profile approach
#for expert finding
#
#(C) 2009 Parin Shah. All Rights Reserved.
#Version 1.00		11/01/2009
##############################################################################


##############################################################################
#	Perl modules used in this program
##############################################################################
use warnings;
use strict;
use CGI qw( :standard );

print header(), start_html( "Detail summaries" );
print '<form id="form1" name="form1" method="post" action="">';
print  '<label>search box';
print  '<input type="text" name="search_box" id="search_box" />';
print  '</label>';
print   '<input name="submit" type="submit" value="submit" />';
print '</form>';

#unlink "hello.txt";
my $var = param("search_box");
print length($var);
if (length($var) != 0) {
	system ("lw_person_profile.pl $var");
	open (IN, "<../../../../Users/PDS/Desktop/rank_based_on_profile.pds") or die "Cannot open the output file...";
	my @output = <IN>;
	my $no_of_expert = scalar(@output);
	if($no_of_expert > 15) {
		print "<p>There are 15 experts found for <b>\"",$var, "\"</b></p>";
	} else {
```

```
		print "<p>There are ",$no_of_expert, " experts found for <b>\"",$var, "\"</b></p>";
	}
	for(my $i = 0; $i <15; $i++) {
		my @temp = split(' ', $output[$i]);
		$temp[0] =~ s/__at__/\@/g;
		$temp[0] =~ s/__dot__/\./g;
		$temp[0] =~ s/__dash__/\-/g;
		print '<a href="mailto:', $temp[0], '">',$temp[0], '</a><br >';
	#	<a href="mailto:manish@simplygraphix.com">Send me an email</a>
	}
	close(IN);
}
print end_html();
```

REFERENCES

[1] Y. Fu, R. Xiang, M. Zhang, Y. Liu, S. Ma, ***"A PDD-based Searching Approeach for Expert Finding In Intranet Information Management".*** The third Asia Information Retrieval Symposium, AIRS06, 2006,

[2] L. Page, S. Brin, R. Motwani, T. Winograd, ***"The pagerank citation ranking:Bringing order to the web"*** . Tech. Rep. Computer Systems Laboratory, Stanford University,Stanford, CA. 1998.

[3] H. Chen, H. Shen, J. Xiong, S. Tan, X. Cheng, ***"Social Network Structure behind the Mailing Lists: ICT-IIIS at TREC 2006 Expert Finding Track"***. The Fourteenth Text REtrieval Conference Proceedings (TREC 2006) , 2006.

[4] N Japkowicz and S. Stephen. *"**The class imbalance problem: A systematic study".*** Intelligent Data Analysis, 2002 - IOS Press

[5] Susan Feldman ***"The high cost of not finding information"*** In Content, Document and Knowledge Management KMWorld-Volume 13, Issue 3, March 2004

[6] D. Mattox, ***"Expert Finder"***, The Edge: The MITRE Advanced Technology Newsletter, vol. 2, no. 1, 1998. (at http://www.mitre.org/pubs/edge/June_98).

[7] D. W. McDonald and M. S. Ackerman, ***"Just Talk to Me: A Field Study of Expertise Location",*** Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work (CSCW'98), Seattle, WA: ACM Press, pp. 315-324, 1998.

[8] A. S. Vivacqua, ***"Agents for Expertise Location"***. Proceedings of the AAAI Spring Symposium on Intelligent Agents in Cyberspace, Stanford, CA, pp. 9 - 13, March 1999.

[9] T. H. Davenport and L. Prusak, ***"Working Knowledge : How Organizations Manage What They Know"***. Boston, MA: Harvard Business School Press, 1998.

[10] D. Yimam. Expert Finding systems for organizations: Domain analysis and the demoir approach. In **"*Beyond Knowledge Management: Sharing Expertise*"**. MIT Press, 2000.

[11] H. Kautz, B. Selman and A. Milewski, ***"Agent Amplified Communication"***. In The Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96), Portland, OR, 1996, pp. 3 – 9

[12] Nick Craswell, David Hawking, Anne-Marie Vercoustre,Peter Wilkins, ***"P@NOPTIC Expert: Searching for Experts not just for Documents"***, CSIRO, Australia 2003

[13] D. Mattox, M. Maybury and D. Morey, ***"Enterprise Expert and Knowledge Discovery".*** In Proceedings of the 8 International Conference on Human-Computer Interaction (HCI International'99), Munich, Germany, pp. 303-307, August 1999

[14] ***"Collaborative Virtual Environments for Analysis and Decision Support"*** by Mark Maybury, Information Technology Division, The MITRE Corporation

[15] *"**Expertise identification using email communications**"* published in November 2003 for the Proceedings of the twelfth ACM conference on Conference on information and knowledge management

[16] "**EOS: expertise oriented search using social networks**" published in May 2007 for the Proceedings of the 16th international conference on World Wide Web