International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 8, August - 2014

User Interaction using Experttop Analysis

Sushant Patil
M.E. Student in Computer Engineering Dept.
St. Francis Institute of Technology,
Mumbai University,
Mumbai- 400103, India

Nazneen Ansari
Assistant Professor,
Department of Information Technology
St. Francis Institute of Technology,
Mumbai University
Mumbai- 400103, India

***Abstract*—To look for expertise, continuous interaction with expert systems is necessary. In many of these systems, it is complicated for a user to decide whether the answer obtained is from a reliable and authenticated source. To overcome this issue, this study provides a new ranking algorithm, i.e., "ExpertTop" algorithm, which allows users to obtain accurate answers from reliable and authenticated sources. ExpertTop algorithm ranks the system experts based on experience and feedback provided through scores by the users and the co-experts within the system. For classifying the user's query, Naïve Bayesian algorithm is used. ExpertTop algorithm will assist users in obtaining results from reliable and authenticated sources, which will provide them satisfactory experience.**

***Keywords*—Human Provided Services, ExpertTop algorithm, Naïve Bayesian algorithm, stop words, stem words**

I. INTRODUCTION

Web services have emerged as a mixture of compassable systems. With the launch of these services, the World Wide Web is changing from a data warehouse to an environment where applications can be automatically invoked by web users or other applications. Web service [1] is a service through which computers can communicate with each other over the web using hypertext transfer protocol and other universally supported protocols. A web service is a piece of information that makes itself available over the internet. Web services have made it possible to publish, locate, and invoke a transition or progress across the web.

Human provided services (HPS) [2][3] enables flexible interaction in service-oriented systems. It allows users to publish their competences and skills as a service. Using HPS, users are able to define and provide services for different associations. HPS permits users to control their interactions beyond simple message exchanges by defining multiple service interfaces and interaction rules to manage complex interactions. The uniqueness of HPS is that collaborations occur in a service-oriented framework, thus enabling a dynamic mixture of human beings and software services.

1.1 Features of HPS

1.1.1.Capable of defining service:
Anybody can define services and corresponding interfaces, or simply reference or copy an existing interface and reuse or modify it.

1.1.2. Discovery and interaction with users or processes:
Processes and humans actors can determine HPS. HPS streamlines interactions with user-provided services by abstracting from service location and deployment.

1.1.3. Specification of interaction:
Users can specify their interaction protocols. Customized protocols permit interactions to be managed in a given context that is in collaboration through services.

1.1.4. User-centric service publishing:
Embodies the skills to effortlessly publish and interact with services.

Recommendations are usually performed by requesting friends or co-workers who may have faced similar problems earlier. Once the expert seeker has recognized a potential candidate, contact can be established using standard tools such as e-mail, instant messaging, or telephone. People tend to know reputable and trusted experts in small environments, and also what data needs to be exchanged to solve a problem.

The disadvantages of HPS are that people require tremendous knowledge about the skills of co-workers and internal structures of the organization. The physical discovery of an expert turns out to be an intimidating task when the number of people increases, for example, the web or large-scale enterprises. With a huge set of people available, assuming that reputation, skills, and trust between people changes dynamically, discovering experts and support turns out to be a major task that cannot be performed manually. To control the disadvantages of manual discovery, expert systems are used [4].

The remainder of this paper is as follows. In section II, the study describes some related work for experts ranking. In section III, the proposed study and overview of the system is discussed. In section IV, Query classification is explained, followed by details of ExpertTop algorithm in section V. Section VI represents the functionality of the ExpertTop algorithm using the images from Figs. 2 to 9, followed by performance analysis in section VII. Finally, section VIII concludes the study.

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

II. RELATED WORK

In [5], Web Service Human Task [6] and Business Process Execution Language (BPEL) [7] were issued to address the need for human interaction support in service-oriented systems. These standards stipulate the need for human interactions in business process. They specify language to model human interactions, the human task lifecycle, and the generic role model. Role-based models are used to model responsibilities and potential task in business process, while BPEL focuses on top–down modeling of business process.

Iris Eiron et al. [8] elucidated a study about graph-based ranking measures to rank email correspondents as per their expertise degree on an interested subject. The main focus is on the analysis of digraphs whose node corresponds to correspondents, whose edges correspond to the existence of email correspondence between the people corresponding to the nodes they connect and whose edge directions point from the member of the pair whose relative expertise has been estimated to the higher. A new error measure for comparing ranked list is introduced. The comparison method of ranking algorithm is based on applying them to graphs where the ground truth ranking is known, and then assessing how well the algorithm output agrees with the true ranking. Real data is used to perform the analysis of graphs, in which edges represent email exchange. The use of text classification to determine the direction of the edges may further degrade the graph quality by assigning incorrect directions and/or weights to the edge.

Jun et al. [9] analyzed a Java-based forum—a large online help seeking community using social network analysis methods. They tested a set of network-based ranking algorithms, including PageRank and hyperlink-induced topic search (HITS), on this large size social network in order to identify users with higher expertise. Simulation was used to identify a small number of simple simulation rules governing the question–answer dynamic in the network. Expertise Rank (ER) algorithm is proposed to generate a measure that not only considers how many people one helped but also whom he/she helped. The perception behind this algorithm is that if B is able to answer A's question and C is able to answer B's question, then C's ER should be increased not just because it was able to answer a question but because it was able to answer someone else's question who himself/herself had some expertise. Therefore, ER provides expertise scores through the question–answer network. The ER of user A is represented as follows:

$$ER(A) = (1-d) + d\left(\frac{ER(U1)}{C(U1)} + \cdots + \frac{ER(Un)}{C(Un)}\right) \ldots\ldots (1)$$

where C(Ui) is defined as the total number of users helping U1 and d is a damping factor that can be set between 0 and 1.

Daniel et al. [10] proposed an Expert Web comprising connected experts, which provide help and support in a service-oriented manner. The Expert Web members are humans, such as company employees, rendering help as online support services or in some cases help is provided by software-based services. For trust emergence, the focus is on social trust to support and guide delegations of requests. In contrast to a common security perspective, social trust refers to the flexible interpretation of previous collaboration behavior [11] and the similarity of dynamically adapting interests. Considering social trust is necessary to guide human interactions, the concept of hubs and authorities [12] was used to rank web pages in search queries using HITS algorithm. The basic approach of ExpertHITS algorithm is to use a metric to calculate the overlap of two sets A and B—a straightforward way to define overlap similarity. An algorithm is presented for matching preferences through calculating overlap similarities of sets of properties. These preferences have impact on the matching of skill properties on lower levels. As mentioned earlier, all nodes in the skill tree that do not have successor nodes are called leaf nodes. For simplicity, we do not consider unbalanced trees or complicated branching structures. In other words, ExpertHITS algorithm is an algorithm for matching elements which may have interaction data (RFS-based interactions) and user profiles holding skill information, and it calculates hub and authority scores.

III. PROPOSED WORK

To help users to pursue knowledge from appropriate expert in the area, an ExpertTop algorithm is proposed. This algorithm aims to appropriately rank the experts based on their experience and feedback from users and other experts. In this section, the proposed architecture is discussed.

3.1. System Architecture:

A general architecture is represented in Fig. 1. For simplicity, the proposed architecture is divided into three parts, i.e., admin, expert, and user.

3.1.1. Admin:

The admin is responsible for the working of the system. The admin will deal with all experts who are registered in the system and are available to provide answers to user's questions. Based on the type of the user query, the expert with the highest rank in that category will be selected to provide the answer.

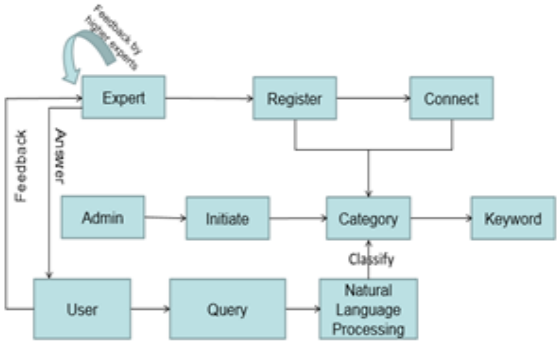

Fig 1. Proposed architecture

3.1.2. Expert:

The expert's task is to provide answers to user's questions. In this block, the expert who has the highest rank will receive a chance to answer the question. In addition, the proposed

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

architecture has the feature for high rank experts to provide feedback to answers of low rank experts.

31.3. User:
User is the knowledge seeker, who waits to receive an appropriate answer from an appropriate expert. When the answer is received from the expert, the user can provide feedback about the answer, i.e., whether the user is satisfied or not. Based on the feedback, the expert rank is decided.

The proposed architecture provides mechanism for other experts who are highly ranked than the current expert to provide their feedback to its answer. For appropriate query classification, the following steps of natural language processing (NLP) are used. (1) Stop word removal—this includes the removal of words, which do not contain weightage and (2) word stemming—the conversion of all word forms into base word, for example, reached and reaching will be converted to reach. This is followed by Naïve Bayesian algorithm [13], which provides the most accurate result compared to other text classification algorithms.

IV. QUERY CLASSIFICATION

For classifying a query send by a user, three steps are followed, i.e., stop word removal, word stemming, and Naïve Bayesian algorithm. The first two steps are based on NLP.

4.1. Stop Word Removal:
It is the process of eliminating words, which do not possess any weightage. The principal use is to avoid the use of common words [14]. Stop words are usually words such as "to," "I," "has," "the," "be," and "or." Eliminating stop words helps in decreasing both the index and the query size.

4.2. Word Stemming:
Stemming is a process for reducing words to their base or root form, usually written in a word form [15]. The stemming process is useful in search engines for query expansion, indexing, and NLP. For example, words such as reached and reaching will be converted to its base word, i.e., reach.

4.3. Naïve Bayesian Algorithm:
The Naïve Bayesian algorithm is a simple probabilistic classifier based on applying Bayes Theorem [16]. It represents a supervised learning method. This algorithm is simple, easy to implement, and has a superior performance. The Naïve Bayesian algorithm is used for text classification, spam filtering, hybrid recommendation, etc.

The Naïve Bayesian algorithm is represented as follows:

$$P(h/D) = \frac{P(D/h)\,P(h)}{P(D)} \quad \text{.....(2)}$$

Where,
P(h): prior probability of hypothesis h,
P(D): prior probability of training data D,
P(h/D): probability of h given D, and
P(D/h): probability of D given h.

For classification, maximum probability is also calculated. This can be understood with the example in [16]. According to [13], Naïve Bayesian algorithm provides the most accurate results as compared to other text classification algorithms.

V. EXPERTTOP ALGORITHM

To select an appropriate expert who answers the query, ExpertTop algorithm is proposed. This algorithm operates on the user's feedback as well as the higher ranked expert's feedback than the current expert. The ExpertTop algorithm is based on the following factors:

5.1. Expert Experience:
This factor focuses on the expert's experience in a particular domain, which plays an important role for selecting an appropriate expert for replying the query. This factor will be considered when the expert registers itself into the system. The expert with highest experience will be rated as ten, whereas the expert with lowest experience will be rated as one, as shown in Table 1. The other experts present in the system will be rated between one and ten. The rank of the experts based on the years of experience is calculated as follows:

$$\text{Rank of Expert (Experience)} = \frac{\text{Experience} * 10}{\text{Maximum Experience}}$$

Where,
Experience = years of experience of an individual expert
Maximum Experience = Maximum experience of an expert in a particular category

Table 1. Expert experience

Expert	Experience	Range (0 – 10)
1	12	Max(10)
2	9	
3	6	Min(1)
4	10	

5.2. Expert Social Connect:
This factor focuses on the expert's social life, i.e., how well the expert is socially connected. The ratio is calculated on friends that the expert has in the specific domain and the number of friends it does not have in that domain. Next, the ratio of these two values is calculated, as shown in Table 2. The calculated value determines the expert weightage in that domain. The weightage is calculated as follows:

$$\text{Rank on Friends} = \frac{\text{Friends in Domain}}{\text{Friends not in Domain}}$$

Where,
Friends in Domain = friends of expert in system with respect to that domain.
Friends not in Domain = No friends of experts in that domain.

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Table 2. Expert social connect

Expert	Specific Domain	Other Domain	Ratio
1	7	5	7 : 5
2	8	3	8 : 3
3	4	6	4 : 6
4	6	7	6 :7

5.3. Feedback from User:

The ExpertTop algorithm functions mainly on the user's feedback. Once the answer is received by the user from the expert, if the user is satisfied by the answer, then the user provides a positive feedback, else the user provides a negative feedback to the current expert. Based on these feedbacks, the expert ranks are adjusted.

5.4. Up/Down Vote from other Experts:

This factor allows other experts to check whether the answer provided by the current expert is appropriate or not. It may happen that there is a delay for the feedback from the user. Therefore, the vote given by the expert who is more experienced or highly ranked than the current expert affects the rank of the current expert.

Moreover, the expression to calculate the "Rank of the Expert" in the system is represented as follows:

$$\text{Rank of the Expert} = (0.2 * \text{Expert Experience}) + (0.3 * \text{Rank on Friends}) + (0.5 * \text{Feedback})$$

VI. RESULTS

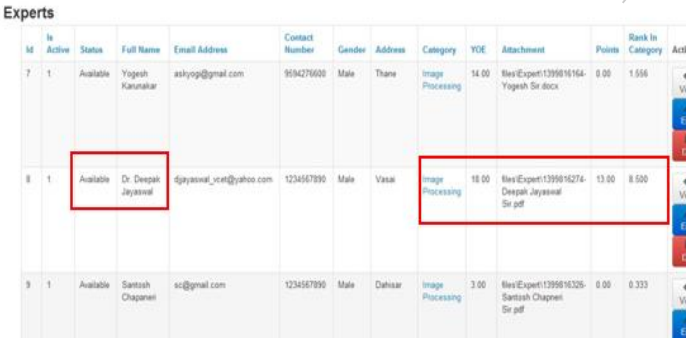

Fig 2. Expert's status and rank before firing the query showing "8.50"

Fig 3. Query fired by a user "Explain Morphology"

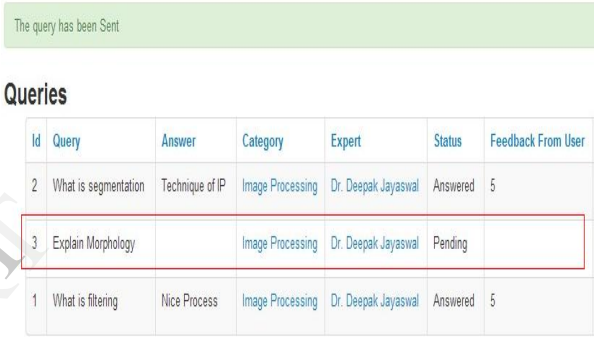

Fig 4. Query status shown "pending" at the expert end

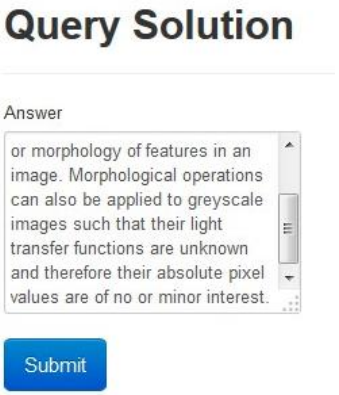

Fig 5. Query answered at the expert end

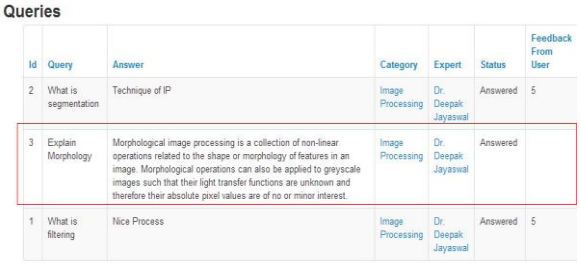

Fig 6. Query answer status "Answered" at the user end

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

Fig 7. Provide feedback at the user end

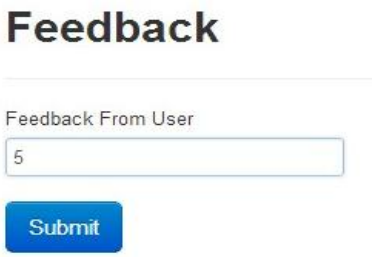

Fig 8. Feedback provided at the user end

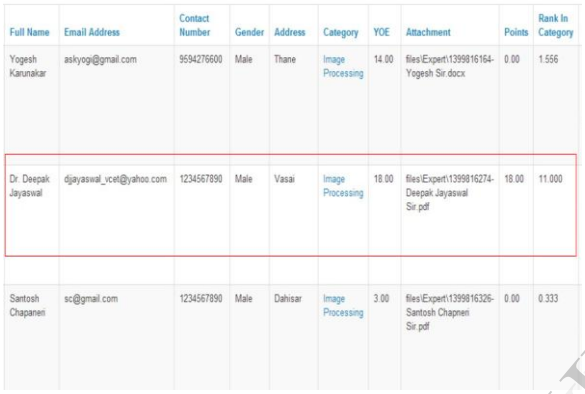

Fig 9. Change in rank of expert after the feedback provided at the user end showing "11.00"

VII. PERFORMANCE ANALYSIS

For evaluating the system, the analysis is performed based on the following parameters.

7.1. Response Time:

This parameter emphasizes on how quickly the experts are able to reply the user's questions. The ExpertTop algorithm helps to provide a faster response to the expert.

7.2. Speed of Categorization:

This parameter emphasizes on how rapidly the system is able to categorize the query in an appropriate domain. Moreover, it considers how appropriately the experts can be categorized based on the ExpertTop algorithm.

VIII. CONCLUSION

We believe that through this study, users will always be able to acquire the appropriate and updated knowledge from authorized experts. The system is made flexible; due to this, the users can provide feedback about the answers or express their views about the obtained results and also rate the experts. Thus, the experts can be judged appropriately based on the user's feedback. If an expert is not able to answer appropriately to a query put forth by the user, then the admin can block the expert for further sessions. ExpertTop algorithm will efficiently rank the experts as well as categorize them accurately. In addition, higher ranked experts are made eligible to rank the lower ranked experts so that the efficiency of the lower ranked experts can be monitored. The system categorizes a given query using stem and stop words. Furthermore, the ExpertTop algorithm operates efficiently and provides optimum and high quality results to recognize reliable and authorized Experts, thus providing the end user desirable, efficient, reliable, and time saving results.

REFERENCES

[1] D. Jordan, *et al.*, "Web Services Business Process Execution Language Version 2.0,"*OASIS standard 11*, 2007.

[2] D. Schall, *et al.*, "Unifying Human and Software Services in Web-Scale Collaborations," *Internet Computing, IEEE*, vol. 12, pp. 62-68, 2008.

[3] D. Schall, "Human Interactions in Mixed Systems-Architecture, Protocols, and Algorithms," *Unpublished Ph.D. thesis,Vienna University of Technology*, 2009.

[4] D. C. Brabham, "Crowdsourcing as a Model for Problem Solving an Introduction and Cases,"*Convergence: the international journal of research into new media technologies*, vol. 14, pp. 75-90, 2008.

[5] F. Leyman, "Workflow-based Coordination and Cooperation in a Service World," In *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, Springer Berlin Heidelberg, pp. 2-16, 2006.

[6] A. Agarwal, *et al.*, "Web Services Human Task (WS-Human Task)," *Version 1.0.Technical report, Active Endpoints Inc., Adobe Systems Inc., BEA Systems Inc., IBM Corporation, Oracle Inc., and SAP AG*, 2007.

[7] A. Agarwal, *et al.*, "WS-BPEL Extension for People (BPEL4People)," *Version 1.0*, 2007.

[8] B. Dom, *et al.*, "Graph-based Ranking Algorithms for E-mail Expertise Analysis," *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, ACM*, pp. 42-48, 2003.

[9] J. Zhang, *et al.*, "Expertise Networks in Online Communities: Structure and Algorithms," In *Proceedings of the 16th international conference on World Wide Web, ACM*, pp. 221-230, 2007.

[10] D. Schall, *et al.*, "Expert Discovery and Interactions in Mixed Service-Oriented Systems," *Services Computing, IEEE Transactions on*, vol. 5, pp. 233-245, 2012.

[11] F. Skopik, *et al.*, "Modeling and Mining of Dynamic Trust in Complex Service-Oriented Systems," *Information Systems*, vol. 35, pp. 735-757, 2010.

[12] J. M. Kleinberg, "Authoritative Sources in a Hyperlinked Environment,"*Journal of the ACM (JACM)*, vol. 46, pp. 604- 632, 1999.

[13] R.M. Rashedur and F. Afroz, "Comparison of Various Classification Techniques using Different Data Mining Tools for Database Diagnosis," *Journal of Software Engineering and Applications*, vol. 6, pp. 85-97, 2013.

[14] Z. Ceska and C. Fox, "The Influence of Text Pre-processing on Plagiarism Detection," *Association for Computational Linguistics*, 2011.

[15] T. M. T. Sembok "Word Stemming Algorithms and Retrieval Effectiveness in Malay and Arabic Documents Retrieval Systems."*Proceeding of World Academy of Science, Engineering and Technology*, vol. 10, 2005.

[16] Naïve Bayes Classification, http://software.ucv.ro/ ~cmihaescu/ro/teaching/ AIR/docs/Lab4-NaiveBayes.pdf

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

SushantPatil*received his B.E. degree in May 2012 in the field of* ***Information Technology*** *from* ***the University of Mumbai****. Currently, he is pursuing his M.E. degree from* ***the University of Mumbai*** *and also works as a Lecturer at St. Francis institute of Technology. His areas of interest include database system and data mining.*

Nazneen Ansari *is currently working as an assistant professor at the department of* ***Information Technology****, St. Francis Institute of Technology Mumbai. Her areas of interests include Artificial Intelligence, Network Security, Database systems, Datamining, Game Programming and Distributed Computing.*

IJERT

(This work is licensed under a Creative Commons Attribution 4.0 International License.)