Proficient Information Retrieval Using Trust Based Search On Expert And Knowledge Users Query Formulation System

Sridharan, K. and Chitra, M.

1Associate Professor, Department of Information Technology, Panimalar Engineering College, Chennai, Tamilnadu, India.
2Professor, Department of Information Technology, Sona College of Technology, Salem, Tamilnadu, India.

ABSTRACT

Due to increase in complexity of services there is required for dynamic interaction models. In two sided service-oriented systems, both human provided services and software services are combined together to provide a single service. For a two sided service system to work properly we need a context-sensitive trust based on both content and expert to our modified trust algorithm which process exact skill matching and retrieval of Information based on proper content rank. Our contribution to this system is this new modified trust algorithm which process to evaluate trust based on the expert rank and their content quality of their resources provided. We also make the system available for public use and available for multi sided systems, hence effectively reduces complexity in combining human interaction system and software services. It provides superior exploration designed for knowledgeable users.

INTRODUCTION

Web services already play an important role in fulfilling organizations’ business objectives. Every day more business processes are opting for an open web based platform and web services for providing their services. We utilize Human provided Services thus enabling flexible interactions in service-oriented systems. There are two ways to search for solutions.

- We can manually discover an expert by asking people for their opinion and manually deciding who is trustworthy.
- We can discover an expert from a pool of experts in an expert hub.

However, these options are not updated regularly since they don’t take into account the learning curve of an expert. Thus we provide a way to dynamically rate experts according to metric values and the new trust algorithm.

An expert hub is a collection of experts with the same interests and qualifications. Each time an expert provides a solution or decides to delegate their score is updated.

Architecture:

The Expert Seeker will serve to search for expert based on their best known knowledge like behaviors and testimonial track record, challenges, cases, journals, news article, personal litigation history, professional history such as employment teaching research, educational history, publications, criminal history, expert declarations, deposition summaries, Associations and memberships summaries and their performance updated dynamically for answering and identifying the solutions for the given query. The preferences are sensitive to context and calculated at the time of choice.

Related work:

Recently, trust in social environments and service oriented systems with HPS has become a very important research area. Most important software vendors have been working on standards addressing the require of human interaction support in service-oriented systems. The finding and communications in mixed service oriented systems comprising HPS and software-based services. Discovering support naturally requires the related expert / knowledge seeker to start an investigation for an expert by asking other people for their view or to provide recommendations; who is able to help and who is trustworthy. Recommendations are typically performed by asking friends or colleagues who may have faced related problems in the past. The drawbacks are that public need extensive knowledge about the skills of colleagues and internal structures.

Corresponding Author: Sridharan, K., Associate Professor, Department of Information Technology, Panimalar Engineering College, Chennai, Tamilnadu, India.
E-mail: sridharank.p@gmail.com.
of the organization. The manual discovery of an expert becomes a very difficult in the Web or large-scale enterprises and given that reputation, skills, and trust among people changes dynamically, discovering experts and support becomes a major undertaking that cannot be performed any longer in a manual way also.

Experts may also delegate RFSs to other experts in the network, for example when they are overfull or not able to provide fulfilling responses. Following this way, not only users of the expert network establish trust in experts, but also trust relationships between experts come out based on their raw data given but verification done in manual.

Fig. 1: Proposed System Architecture.

Assignment is important in flexible, interaction-based systems because specialist hubs will attract many RFSs over time, thus presenting bottlenecks in terms of processing or delegating RFSs. On the other side, being a hub in the Expert Web also means that a person knows many other experts in similar fields of interest. We argue that the likelihood of being able to delegate RFSs to other experts very much increases depending on the hubness of a person arising from being a part in expert areas. The major challenge in this scenario is that hubness needs to be calculated on demand based on a given query at dynamically. A query determines the context specified as the set of relevant skills. The following are some of the issue consider and enhanced in proposed of my works.

• The user is incapable to trust the “experts” opinion since ranking is based on expert evaluation and not calculated more dynamically on demand.

• Graphplan looks for the actions and previous states from which goals can be reached, pruning as many of them as possible, but this is not possible for a new problem without any current solution.

• EMA gives more importance to recent ratings and has not been applied to real collaboration environments to capture more realistic interaction data.

• ExpertHITS in human-centric and social collaborations are not properly adopted two-sided markets in mixed oriented services system.

Therefore we are consider this problem and give the solution. Applied to enterprise scenarios, such a network of experts, spanning various organizational units, can be consulted for efficient discovery of available support. Ranking changes dynamically according to an experts success rate.

This model also relies on the availability of a workload of queries spanning all attributes and values to establish a score for a tuple. However, a drawback of such a workload is that in the context of Web databases, user queries are restricted to a subset of the attributes that are displayed in the results. In such a setting, the workload will fail to capture user preferences towards those attributes and values that cannot be specified in the query. In contrast, we capture these preferences via users’ browsing choices in a query- and user-dependent setting. One method of keyword extraction are based on keyword matching with basic TF-IDF weighting (Joachims and Thorsten, 1996). The additional method is based on using context information to develop keyword extraction (Jasmeen and Gupta, 2010). There are more techniques developed to use local and global context in keyword extraction (Zhang, 2006; Jasmeen and Gupta, 2010; Bao and Zhen, 2012). The other techniques used to improve information retrieval uses concept of semantic analysis such as ontology based similarity measures (Fernandez and Miriam, 2011; Soner, 2012). In these approaches the ontology information is used to discover similarity between words and find words still if the exact match is not available.

Another ways in which semantic information is extracted is using Wordnet libraries. Wordnet based approaches have used concepts such as relatedness of words for information retrieval (Feng, 2011).

The current World Wide Web is the greatest global database that lacks the existence of a semantic structure and therefore it becomes tricky for the machine to understand the information provided by the user in the form of search strings (G. Madhu, et al., 2011). Hakia (D. Tumor et al., 2009) is a general purpose semantic search engine that search structured text like Wikipedia.

Expertise model:

In this section, we are discovering experts based on their skills which are given by an expert seeker in the form of a personalized query. Before an expert can provide the services he has to be rated as a trusted expert. When registering to be a knowledge worker, the experts trust has to be initiated. Then based on their skills they are given a hub score. For each hub
an authority is decided again based on ranks. These ranks are calculated dynamically based on success and failure. A skill model is also proposed as a classification system.

- Enhancement of Trust Emergence
- Custom-Made Expert Queries
- Modified Expertise Model

**Enhancement of Trust Emergence:**

These are rankings given by every expert that has interacted with the given expert. Social aspects need to be considered and require dynamic interaction models. In this system, we focus on social trust to support for judge the expert details and their skills, guide assignment of requests if the related experts are overload on demand and also guide the knowledge seeker to the appropriate expert. In contrast to the security perspective, social trust refers to flexible analysis of previous relationship activities and the correspondence of dynamically adapting interests. Especially in mutual environments where users are showing to higher risks than in common shared network scenarios, and where business is at stake, considering public trust is essential to effectively guide human interactions.

Here we bring in the new trust algorithm which will be explained in later sessions, which can be very helpful for the humans to be guided to the right expert.

The Initial trust is done in the form of verification, which is being done by the administrator. Administrator verifies the Designation, qualification, experience and their related certificates.

**Custom-Made Expert Queries:**

Delegation is significant in a system such as ours that is both flexible and interactive. Since these kinds of expert Hubs will attract many RFSs, we need to make sure there are no restricted access during processing of RFSs. And also successful delegations are based on the probability that the other expert has similar interest. This is called a context sensitive adaptive system. First basic matching is performed based on the query to find the skills and then the experts are discovered based on the information. We are using the category of skills during outline expert discovery.

**Modified Expertise Model:**

Our proposed expertise model is based on the ACM Computing Classification System. This simple model is sufficient to serve as a basic classification scheme for skills in the computer science domain. Our ranking model can be extended to other domains as well by using different taxonomies. The basic idea of our approach is to define different weights for each level in the tree. The topmost level (the root node) has the lowest weight since top levels in the skill tree denote broad areas of expertise. The weights increase depending on the tree depth because lower levels contain fine-grained skill and expertise information (specialism). We define the set of levels $L = \{L_0, L_1, \ldots; L_n\}$. All nodes in the skill tree that do not have successor nodes are called leaf nodes. A subset of the tree may be selected by a query $Q$ to discover experts. Thus, the provided query needs to be matched against user profiles to determine how well users match the demanded set of skills. Each node in the tree is called Skill property.

We introduce query preferences enabling the expert seeker to express strong or weak matching preferences of skill properties and optional ("nice to have") properties. A strong preference might denote that the expert seeker requires the user to have certain skill properties, whereas weak preferences would express that the expert should have as many skill properties. Optional means that higher preferences are given to those experts that have a
higher degree of similarity with a set of optional skill properties.

**Expert discovery:**

Before an expert can be approached for a problem we have to first discover the appropriate expert with the right skill necessary. A skill matching algorithm is required to match the skills required and the skills of an expert. Expert hubs need to be discovered and this is influenced by social trust and rating mechanisms. Discovering hubs, Delegation actions when the expert fails to provide the answer, Trust based delegation patterns, rating procedure for rating experts, trust updates based on interactions from interaction metrics.

- Skill Matching Algorithm
- Discovery of Experts
- Metrics Calculation

---

**Skill Matching Algorithm:**

In this work, we present an algorithm supporting the notion of strong, weak, and optional matching preferences through alternate approaches for calculating overlap similarities of sets of properties. We use an algorithm called topic tree matching algorithm.

**Discovery of Experts:**

At this time we present our new version of expert discovery algorithm that is influenced by common trust and ranking mechanisms. Our algorithm accounts for context information and weighted associations between actors. Context is utilized by considering the relations of experts in different scopes. Thus, the goal of our algorithm is to find proper, trustworthy and valid experts with respect to contextual information after verified and top rated.

The figure shows how the expert is selected according to their skills and also their rankings due to their previous interactions with other users.

---

![Data Flow Diagram of Expert Discovery](image)

*Fig. 3: Data Flow Diagram of Expert Discovery.*

![Logical Design of Skill Matching](image)

*Fig. 4: Logical Design of Skill Matching.*
metrics calculation:

It is a combination of few set of values and their formulation which is utilized for trust inference, below we have the metrics table which has the set of metrics considered, there range and the calculation description.

Table 1: Metrics Table

<table>
<thead>
<tr>
<th>Metrics Name</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessibility</td>
<td>[0, 100]</td>
<td>Ratio of accepted to received RFSs</td>
</tr>
<tr>
<td>Response Time</td>
<td>[0, 0.97]</td>
<td>Average response time in hours</td>
</tr>
<tr>
<td></td>
<td>[0, 0.49]</td>
<td>Between seeker and expert</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Between Expert and delegated expert</td>
</tr>
<tr>
<td>Success Rate</td>
<td>[0, 100]</td>
<td>Amount of successfully processed RFSs</td>
</tr>
<tr>
<td>Experience</td>
<td>[0, ∞]</td>
<td>Number of RFSs processed</td>
</tr>
<tr>
<td>RFS reciprocity</td>
<td>[-1, 1]</td>
<td>Ratio of processed to sent RFSs</td>
</tr>
<tr>
<td>Reward</td>
<td>[0, 0.5]</td>
<td>Manually assigned score</td>
</tr>
</tbody>
</table>

The response time is calculated as the duration between sending (or delegating) a request to a service and receiving the corresponding response, averaged over all processed RFSs.

\[
\tau = \frac{\sum_{rfs \in RFS} (t_{received}(rfs) \cdot t_{send}(rfs))}{|RFS|}
\]  

(1)
An RFS is considered successfully processed (sRFS) if leading to a result before a predefined deadline, otherwise it fails (fRFS).

\[
\text{sr}_Q = \frac{\text{num}(sRFS)}{\text{num}(sRFS) + \text{num}(fRFS)}
\]

Delegation Model:

Delegation is the most important aspect of our system and requires a real time connection between expert seekers and the entire hub. When a seeker has found the right expert, he sends the RFS request for a problem.

The receiving expert tries to solve the problem. If the expert cannot solve it, then he has the choice of rejecting the request or delegating it with other experts in the hub.

If the receiving expert has not answered the query within a given time limit then the query is converted to “failed to response” and the seeker is announced about its failure. And he is requested to choose another expert.

In the existing system there is a use of triadic delegation pattern but here we do not use this pattern, this is done in order to overcome conflict over the response to the query.

Entrustment Action:

If some of the experts are known to answer the query but do not have the amount of time to it, so we have provided the delegation services for the expert to delegate some other expert or choose automatically second highest rank of experts in a related domain if they are not delegated.

Fig. 7: Data Flow Diagram of Delegation Model.

User u receives an RFS issued towards the Expert Web. Since you represent the hub expert, you may decide to delegate the request to any number of users of its neighbor v; w; y; z, which can be discovered through knowing relations. In our Expert Web application scenario, there is a bidirectional relation between users. A relation becomes active if both users acknowledge that they are connected to each other (v knows u and u knows v), a simple yet effective mechanism to support growth in social networks while preserving user control. Notice, knows relations do not contain context related information such as tags. The context of interactions is derived from delegated RFSs. Figure 4 shows how the expert receives the request and Figure 6 shows how the expert delegates it to another expert.

My Proposed methodology of new trust algorithm:

The trust algorithm is processed right after the expert is being selected. This algorithm is to improve the trust of the query response seeker over the expert who is going to answer to the requested query.

Here we provide a new algorithm for the improvement of the trust, it mainly consists of 4 important functions they are as follows:

Skill Level Calculation:

It is the process which is used to calculate the level of expert skill of the particular domain, the value N ranges between 0-25 (approximately) is returned. The skill levels are valued of given below

<table>
<thead>
<tr>
<th>Person</th>
<th>Type of domain</th>
<th>N=Value awarded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expert</td>
<td>Main</td>
<td>25</td>
</tr>
<tr>
<td>Expert</td>
<td>Optional</td>
<td>20</td>
</tr>
<tr>
<td>Expert</td>
<td>Basic-Main</td>
<td>15</td>
</tr>
<tr>
<td>Expert</td>
<td>Basic-Optional</td>
<td>10</td>
</tr>
</tbody>
</table>
Personalized Query Comparison:
It is the sub process which is used mainly to provide the priority level of the query context for the expert in his respective domain with all related information. Segment a page into different sections to find sub-topics/concepts only in the appropriate sections. For example, for a given query finding definition of each concept using syntactic patterns as follows.

{is | are} [adverb] {called | known as | defined as} [concept]
{concept} {refer(s) to | satisfy(ies)} …
{concept} {is | are} [determiner] …
{concept} {is | are} [adverb] [being used to | used to | referred to | employed to | defined as | formalized as | described as | concerned with | called}

Algorithm:
1. Compare query content with the context based on my pattern proposed of the expert user domain
2. Generate the priority level based on the content quality of the query by finding the position of query
3. For all content retrieved based on it calculate the value of priority out of 25 (approximate number of link displayed in a page)

\[ P = \frac{N \times 25}{TC} \]  
\( P \): Priority value of a query
\( N \): Priority value of selected category
\( TC \): Total no of categories

4. Repeat it until all the documents are verified and return the calculated value.

Document Rate Calculation:
It is the dynamic process automatically executed by specified period and which is used to find the rating value of a query of an expert and generate the value ranging 0–25

Algorithm:
1. Get the calculated query priority value \( P \).
2. Calculate the value of rating for the document content out of \( f_v \) by the expert system.
   \[ dr = \frac{\sum ev \times f_v}{n} \]  
\( dr \): Document rank
\( ev \): Expert document value
\( f_v \): Feedback rate by expert (range from 0 [low quality content] – 5 [high quality content])
\( n \): Number of documents
3. Repeat step 2 for all \( P \) and return the value of document rating out of 5.

Expert Rank By Metrics:
It is the one of the automatic process invoked by specific time period and which is used to generate the expert rank by their skill based metrics calculated like domain expert, knowledge expert, etc, it helps to update expert rank periodically.

Algorithm:
1. Select the expert \( E \) from expert list \( L_e \)
2. Calculate updated metrics values of the selected expert \( E \)
3. Rank the expert based on each of their updated metric values
4. Generate expert points for the selected expert \( E \) based on they are ranks
5. For each selected expert \( E \) from \( L_e \), sum the points and calculate the updated expert rank
6. Return the value of updated expert rank.

```java
protected void expedrop_SelectedIndexChanged(object sender, EventArgs e)
{
    double cc = 0.0;
    string ss = expedrop_SelectedItem.ToString();
    Session["expe"] = ss.ToString();
    int sl = expedrop_SelectedValue.Length;
    string ss1 = ss.Substring(sl - 4, 4);
    cc = selectcall(ss);
    cc = cc + querycall(se, ss1);
    cc = cc + ratecall(se, catenum.Text.ToString());
    cc = cc + rankcall(se);
}
```

Listing 1:
Sample for New Trust Algorithm Invocation Model

RESULTS AND DISCUSSIONS
In this section we evaluate the performance of the proposed approach on two academic search engines namely, Google Scholar (GS) and Microsoft academic search (MAS). We want to compare and discuss the performance using different search engines and propose what kind of search engines are most suited for dataset extraction in real application. In order to evaluate the proposed approach we have used real dataset. The dataset was constructed as described below.
**Dataset Description:**

In order to evaluate the performance of the Trusted DataExtract algorithm proposed in this work, the following dataset was constructed. We have selected 50 research papers from various computer science conferences such as KDD, ICDM, WWW published between 2012 till 2014. The collection of these 50 papers is the corpus on which the Trusted Data Extract algorithm is tested. In order to construct this dataset we considered only full length papers and excluded any workshop or poster papers from the corpus. Also, in order to reduce the search space and increase the content quality we have removed papers which did not experimented on real datasets. The test dataset consists of dataset names associated with each paper.

**Evaluation Metrics Used:**

As described in the last section, we have used 50 papers from the corpus C to extract dataset names from these papers. For each paper we get a list of dataset names that were extracted from the paper. We have used the standard evaluation metrics such as precision, recall and F-measure. In the standard information retrieval terminology, these metrics are defined as follows:

- **Precision (P):** The ratio between the number of relevant items in retrieved items and the total number of retrieved items. Items here mean the dataset names. This is computed for each of the test paper $p_i$ and then averaged for all the papers to get an average precision.

- **Recall (R):** The ratio between the number of relevant items in the retrieved items and the total number of relevant items. Recall is computed for each of the test paper $p_i$ and then averaged for all the papers to get an average recall.

- **F-measure (F):** A measure that combines precision and recall is the harmonic mean of precision (P) and recall (R). The F-measure is computed using the average precision and average recall values.

Table I summarizes the performance of the dataset extraction algorithm using two different search engines namely, GS and MAS in terms of precision, recall and F-measure. The precision, recall and F-measure values in this table are computed for datasets in 25% of the total papers considered for this experiment. As described earlier, the pruning radius ($\lambda$) goes in as the parameter for this algorithm. We determine the optimal value of this radius $\lambda$ by constraint maximization. In order to do this, the original test data (consisting of 50 papers) is divided into two parts. The first part, the training set, consists of 75% of the total papers and the test set consists of the remaining 25% of the papers. Once the parameter $\lambda$ is computed from the training set, we evaluate the algorithm’s performance on the test set. In the experiment we also show the difference in the $\lambda$ obtained from GS and MAS. As shown in Fig. 2, the precision, recall and the F-measure values are computed for different values of $\lambda$ (ranging from 0.1 to 1.0) on the training set. From this figure, we can observe how the precision, recall and F-measure value change as $\lambda$ is increased from 0.1 to 1.0. The precision for small $\lambda$ tend to be as high as 100% because very less datasets were extracted at this radius. Since the recall is low at this radius.

Web is the perfect experimental apparatus. However, there is a lot of confusions when comparing research efforts from different point of view. We implemented as webservces and tested with following features. Trust information is periodically updated to capture dynamically changing interaction preferences and trust relations. More trust over query response by Expert/Normal Candidate with the help of new trust algorithm. The user will have better view over the selected expert / knowledge person by knowing the --% amount the query can be processed by expert. The trust algorithm is processed right after the expert is being selected. Here we use a new algorithm for improvement of the trust, it mainly consists of important functions they are as skill level, verification of expert based on their processed request and responses of related skills other than that qualification. Trust calculation based on their responses and availability of experts, assign priority for the request and expert based on the query submitted. Skill level is the function which is used to calculate the level of expert skill for the particular domain, the value ranges between 0-25. Here we include the expert activation and deactivation feature to activate and deactivate the expert based on time of request and response ratio frequently. If the request and response ratio is less for long period, the expert will deactivated automatically.

These features ensure that our algorithm discovers and validated experts which are well-connected to other experts. The query time for request and response is less with more accuracy.

We consider here the two search engines are yahoo ask and stack overflow for comparison of my proposed. The term web content mining has been used to refer to techniques that include a extensive range of issues. An evaluation process and a computing technique for content quality rating with expert proof were designed in the proposed model.

**Conclusion:**

This system involve the activities to solve emergent problems in distributed collaboration environment. Our proposed approach is based on the Human-Provided Services concept enabling knowledge workers and expert to offer their skills and expertise in service-oriented systems. Trust information is periodically updated and verified to capture dynamically based on the request and response rate of the given query. Expert Rank is computed in an online manner, thereby enabling full personalization at runtime. Existing approaches in personalized expertise mining algorithm typically
perform offline interface analysis and depend mostly on associations between experts for faith appearance and authority scores. This expert interaction system performs analysis online dynamically and simplifies the task of selecting experts for the users and simplifies the response of the experts in normal requests as well as delegations. There is an option for the expert seeker to use either an expert or just an knowledge worker; thereby decreasing the chances of bottleneck of RFSs. Open Web based Human Provided Services are highly useful in product support as well as in other applications. This system is available for public use and can be used in two-sided services environment.

<table>
<thead>
<tr>
<th>Expert Information Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expert Type:</strong> Normal</td>
</tr>
<tr>
<td><strong>Expert ID:</strong> 4</td>
</tr>
<tr>
<td><strong>Expert UserID:</strong> ed</td>
</tr>
<tr>
<td><strong>Expert Name:</strong> ed</td>
</tr>
<tr>
<td><strong>Email ID:</strong> <a href="mailto:e13691@hotmail.com">e13691@hotmail.com</a></td>
</tr>
<tr>
<td><strong>Designation:</strong> Designer</td>
</tr>
<tr>
<td><strong>Experience:</strong> 1 year</td>
</tr>
<tr>
<td><strong>Qualification:</strong></td>
</tr>
<tr>
<td><strong>Domain:</strong> 2.11.4</td>
</tr>
<tr>
<td><strong>Domain (optional):</strong> 0</td>
</tr>
<tr>
<td><strong>Categories:</strong> 1</td>
</tr>
<tr>
<td><strong>Languages:</strong> file management, inheritance, class and objects</td>
</tr>
<tr>
<td><strong>Categories (optional):</strong> 1</td>
</tr>
<tr>
<td><strong>Top Projects:</strong> n/a</td>
</tr>
<tr>
<td><strong>Books written:</strong> n/a</td>
</tr>
<tr>
<td><strong>Certificate 1:</strong></td>
</tr>
<tr>
<td><strong>Certificate 2:</strong></td>
</tr>
<tr>
<td><strong>Photo ID:</strong></td>
</tr>
<tr>
<td><strong>Present Status:</strong> 4/3/2013 8:22:44 PM</td>
</tr>
<tr>
<td>![Activate]</td>
</tr>
</tbody>
</table>

**Fig. 8:** Sample Screen of Expert Information And Its Status Details.

**Advanced Online Expert Services**

<table>
<thead>
<tr>
<th>Expert ID</th>
<th>Expert Name</th>
<th>Type</th>
<th>Category1 (Main)</th>
<th>Category2 (Optional)</th>
<th>Full Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>ed</td>
<td>Normal</td>
<td>2.11.4</td>
<td>0</td>
<td>n/a</td>
</tr>
</tbody>
</table>

**Fig. 9:** Sample Screen of Validation Process.
Fig. 10: Sample Screen of Expert and knowledge worker registration.

Fig. 11: Precision Vs. Recall.

REFERENCES


