Assessing Expertise Awareness in Resolution Networks

Yi Chen[1], Shu Tao[2], Xifeng Yan[3], Nikos Anerousis[2], Qihong Shao[1]

[1] *Computer Science and Engineering, Arizona State University*
Tempe, AZ 85287, USA

[2] *IBM T. J. Watson Research Center*
Hawthorne, NY 10532, USA

[3] *Computer Science Department, University of California*
Santa Barbara, CA 93106, USA

Abstract

Problem resolution is a key issue in the IT service industry. A large service provider handles, on daily basis, thousands of tickets that report various types of problems from its customers. The efficiency of this process highly depends on the effective interactions among various expert groups, in search of the resolver to the reported problem. In fact, ticket transfer decisions reflect the expertise awareness between groups, thus encoding a sophisticated resolution social network.

In this paper, we propose a computational framework to quantitatively assess expertise awareness, i.e., how well a group knows the expertise of others. An accurate assessment of expertise awareness could identify the weakest components in a resolution system. The framework, built on our previously developed resolution engine, is able to calculate the performance difference caused by excluding a node from the network. The difference exposes the awareness of this node to other nodes in the network. To our best knowledge, this is the first study on this problem from a computational perspective. We tested the proposed framework on a large set of real-world problem tickets and validated our discovery by carefully analyzing the tickets that are incorrectly transferred. Experimental results show that our framework can successfully capture groups that do not know others' expertise very well.

1. Introduction

Problem solving is a key issue in IT service industry. Every day, help desks or call centers of service providers such as IBM, AT&T, and Citi, may receive thousands of phone calls and emails from their customers who are seeking technical supports. The reported problems range widely from login failure, application crash, to broken transactions. The problem resolution process is reflected by the life cycle of problem tickets. A ticket is opened as soon as a problem is reported. It is routed among various expert groups until it reaches a group that can solve the problem and close the ticket. Table 1 shows a sample ticket transferred multiple times before it was solved. ⟨A, C, E, F⟩ forms its resolution sequence.

ID	Time	Entry
28120	07-05-14	New Ticket: DB2 failure
28120	07-05-14	Transferred to Group A
28120	07-05-14	Contacted Mary...
28120	07-05-14	Transferred to Group C
28120	07-05-14	Status updated ...
28120	07-05-15	Transferred to Group E
28120	07-05-15	Web service checking
28120	07-05-18	Can not solve the problem.
...	...	...
28120	07-05-18	Transferred to Group F
28120	07-05-22	Resolved

Table 1. A sample ticket

As one can see, the efficiency of problem solving depends on two key factors: (i) the capability of each group to solve a problem in its technical area, and (ii) its awareness of the expertise of other groups in order to make a wise decision to transfer an unsolved ticket. Indeed, the efficiency of problem resolution critically hinges on the efficiency of identifying the problem resolver. It is not uncommon that due to human error or inexperience, a ticket is mistakenly transferred to a group who cannot solve the problem or who cannot distribute it properly. Both situations might lead to a long resolution sequence.

In order to accelerate problem resolution, providing better expertise training and awareness training is the key. Expertise training will increase the capability of solving a ticket in each group, while awareness training will improve the probability of transferring tickets to the right groups. Since it is time-consuming and costly to provide training to all of the groups in the system, it is very important to identify the weakest components in the resolution network. If such information is available, training programs could focus on specific employees to reduce the ticket response time.

While it is easy to calculate the total tickets solved by one group, it is not easy to evaluate the expertise awareness due to the presence of ambiguity. Traditionally group awareness is only attempted by HR representatives or managers using survey, peer reviews, and questionnaires, e.g., using questions like “do you know the skill of group A?”, “How well do you know A’s skill level, choose from 1 to 5”, etc. A survey process is often tedious, time-consuming, and error-prone. It is also very subjective.

In this paper, we approach this problem from a unique angle: we develop a computational framework to objectively and quantitatively assess the awareness of expert groups in problem resolution, without social experiments. This provides a cost-effective alternative to user surveys. *Resolution social network*s are built to model problem solving, where nodes represent expert groups, and edges represent ticket transfer relationships between groups. The transfer relationship actually implies the expertise awareness between two groups.

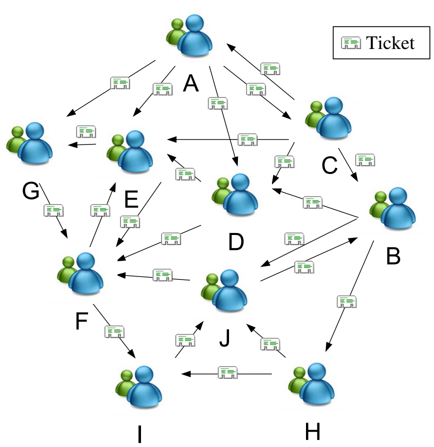

Figure 1. Resolution Social Network

Figure 1 is a sample resolution social network. The relationships in this network are more complicated than traditional social networks studied in social science. For example, expert groups in resolution network may not describe clearly how well they communicate with each other or how much they know others’ skills. Instead, we have to discover the level of expertise awareness by ourselves. This knowledge discovery task is also different from existing social network analysis works, which often focus on topological measures such as degree, closeness, and betweenness [15], [7], [19], [13], [11], to evaluate the importance of a node.

We develop an exclusive framework to evaluate the value of a node E in a resolution social network. By removing E from a network M, we are able to compare the state difference with versus without E. This difference could tell us the value or the importance of E in this network. In this framework, there are several challenging issues. First, should we conduct real-life experiments and examine the effects of removing a group in real ticket resolution? This could lead to prohibitive cost. Second, is it possible to simulate such removals virtually? Third, given a resolution network, how should we measure the change before and after the removal of a node? Is the change significant? In this study, we investigate all these issues and solve them innovatively using a virtual ticket resolution engine.

The significance of our work has two folds. It can not only improve problem ticket resolution, but also provide insights on human interaction in a working environment. By providing appropriate training, weak groups are able to work and communicate more effectively. The technical contributions of this work include:

- We propose an objective and quantitative framework to analyze the interaction of expert groups in a ticket resolution network, and evaluate their awareness of others’ expertise. To our best knowledge, this is the first study on this problem from a computational perspective.
- Ticket data from an IT company was experimented. Case studies show that our algorithm can successfully capture groups with low productivity in a resolution social network.

The remainder of the paper is organized as follows. We first explain the intuition of our exclusive framework in Section 2, followed by a brief introduction on our virtual resolution engine in Section 3. We propose the expertise awareness model in Section 4. In Section 5, we use real world examples to illustrate and verify the discovery of the proposed approach. Related work is presented in Section 6 and Section 7 concludes the paper.

2. An Exclusive Framework

A problem ticket can be represented by a tuple with two components, $(\tau, G_{(k)})$, where τ is the ticket content and $G_{(k)}$ is the routing sequence. Let

$\mathcal{G} = \{g_1, g_2, \ldots, g_n\}$ be the set of all expert groups. The routing sequence of a ticket can be written as $G_{(k)} = \langle g_{(1)}, g_{(2)}, \ldots, g_{(k)} \rangle$ ($g_{(i)} \in \mathcal{G}$), in which a ticket is first issued to $g_{(1)}$, then transferred in the order of $g_{(2)}, g_{(3)}, \ldots, g_{(k)}$. A *step* in $G_{(k)}$ is a ticket transfer from one group to another. A ticket $(\tau, G_{(k)})$ is open if none of the groups in $G_{(k)}$ can resolve it. Correspondingly, a ticket is closed if the last group in $G_{(k)}$, i.e., $g_{(k)}$, solved the problem, and in this case, the routing sequence is called a *resolution sequence*. The efficiency of problem resolution can be measured by the Mean number of Steps To Resolve (MSTR) for m tickets:

$$T = \frac{\sum_{j=1}^{m} |G_j|}{m}. \quad (1)$$

The resolution sequences capture the interaction or expertise awareness among different groups. Intuitively, different expert groups may play different roles in resolving a problem. A group with certain expertise can serve as *the resolver* of a problem ticket. Some groups serve as *transferrer* of problem tickets, who relay the unsolved tickets to other groups. Most of groups usually play these two roles at the same time. For those groups who only serve as transferrer, they are called *distributors*. Distributors do not resolve the problems, but are considered to have good knowledge about the expertise of other groups.

Given a collection of ticket resolution sequences, it is easy to calculate the solving capability of a group, e.g., using the number of tickets resolved by the group. However, it becomes complicated and difficult when one wants to evaluate the transfer effectiveness of a group. When a ticket is transferred from group $g_{(i)}$ to $g_{(j)}$, there are three cases.

Case 1. $g_{(j)}$ is the resolver, i.e., $g_{(i)}$ knows that $g_{(j)}$ is able to answer it correctly, written as $g_{(i)} \twoheadrightarrow g_{(j)}$,

Case 2. $g_{(j)}$ knows how to route the ticket to a right group, i.e., $g_{(i)}$ knows that $g_{(j)}$ is able to transfer it correctly, written as $g_{(i)} \mapsto g_{(j)}$,

Case 3. $g_{(j)}$ is neither a resolver nor the right group to transfer, $g_{(i)}$ mistakenly sent the ticket to $g_{(j)}$. That is, $g_{(i)}$ is not aware that $g_{(j)}$ is unable to handle this ticket, written as $g_{(i)} \not\twoheadrightarrow g_{(j)}$.

For a given ticket, it seems difficult to distinguish Case 2 and Case 3 if $g_{(j)}$ is not a resolver. For example, suppose we have a ticket resolution sequence, $A \rightarrow B \rightarrow C \rightarrow D$. It is hard to distinguish if $A \not\twoheadrightarrow B$ or $A \mapsto B \mapsto C$, etc. Because of this ambiguity, it is hard to assess the effectiveness of a transfer (we use expertise awareness and transfer effectiveness exchangeably) in one ticket. However, for a set of tickets, it becomes possible. This paper will introduce an exclusive framework designed for this task. The rest of this section is going to briefly explain the main idea behind this framework for evaluating transfer effectiveness.

The issue of inefficient ticket solving arises from the lack of knowledge on transferring a ticket to the right group. For example, in Figure 1, group D might have limited knowledge of transferring unsolved tickets to other groups: it always passes unsolved tickets to two groups.

A fundamental question is how to evaluate the "transfer effectiveness" of D to other groups. Assume there is a set of tickets G, in which D is involved as a transferrer and its average MSTR is T_G. For each ticket, one could resubmit it to the system and resolve it without involving D, i.e., D is excluded from the processing of these tickets. Let G' be the new resolution sequences generated by this process. Since D is eliminated, G' could be significantly different from G. We can conclude as follows,

If $T_{G'} < T_G$, then D is not an effective transferrer; or D is not aware of other groups' expertise.

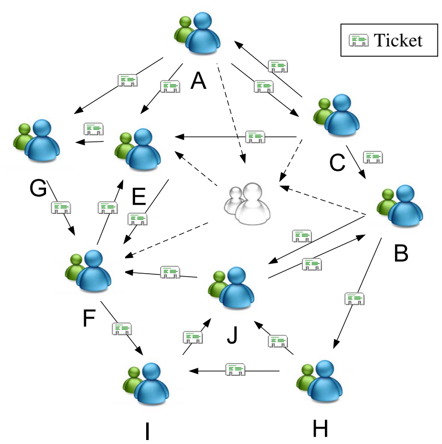

Figure 2. Transfer Effectiveness

Figure 2 illustrates the above idea and tests the system performance by removing D from the resolution social network, for those tickets where D is a transferrer.

The methodology of our framework has been adopted in experimental science: in order to evaluate an element E's impact to the whole system, one could knock out E, and then compare the state of the system in the presence of E with that in the absence of E: The delta reflects the value of E. Adopting this method in a resolution social network, we can evaluate the value of an element (e.g., an expert group, or an interaction edge) by excluding it from the network.

In this framework, we need to calculate the new MSTR based on a modified social network that excludes a node or an edge. Unfortunately, it is impractical to re-execute the previous tickets in a service center by excluding groups or prohibiting interactions. Rather than modifying a real-world social network, we build a virtual network (a ticket routing engine, called VMS, see Section 3) for experiments. This engine could automatically generate a resolution sequence for different tickets.

3. A Ticket Routing Engine

In this section, we will briefly introduce an automatic ticket routing engine, called VMS (Variable-order Multiple active State Search). More details of this applicable model can be found in [14]. The problem of ticket routing recommendation is specified as follows:

Input: A database of historical ticket resolution sequences D_S and an open problem ticket (τ_j, G_j).

Output: The next group that the open ticket should be transferred to.

Objective: To reduce MSTR (Mean number of Steps To Resolve) as much as possible.

3.1. Markov Model

Our method is based on a Markov model. Let each Markov state represents a group, the transition probabilities between these states capture the local routing decisions, i.e. the likelihood of a group to be a transfer target, given the previous groups that have processed the ticket. To build the Markov model, we first extract routing sequences from problem tickets, e.g., sequence $\langle \mathtt{A}, \mathtt{C}, \mathtt{E}, \mathtt{F} \rangle$ from the sample ticket in Table 1. Recall that a process is considered Markovian if at any time point, the probability of the process in the current state is solely dependent on its previous states. Let us use $S_{(k)}$ to denote the set of groups in $G_{(k)}$, i.e. $S_{(k)} = \{g_{(1)}, g_{(2)}, ..., g_{(k)}\}$. The number of historical tickets that share the same set of transfer groups $S_{(k)}$ is denoted as $N(S_{(k)})$. We can estimate the probability of transferring a ticket to g_i given a set of previous transfers $S_{(k)}$, $P(g_i|S_{(k)})$ by

$$P(g_i|S_{(k)}) = \begin{cases} N(\{g_i\} \cup S_{(k)})/N(S_{(k)}) & \text{if } N(S_{(k)}) > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

To determine the "optimal" order of a Markov model, we consider the conditional entropy of the training data and evaluate the entropy of the next group g conditioned on a given set of past groups $S_{(k)}$, which is denoted as $H(g|S_{(k)})$:

$$H(g|S_{(k)}) = -\sum_{S_{(k)} \in \mathcal{G}^k} P(S_{(k)}) \sum_{g \in \mathcal{G}} P(g|S_{(k)}) \log P(g|S_{(k)}). \tag{3}$$

Although increasing Markov order generally leads to better predictability, it also increases the complexity of our model, especially when we apply it to online ticket transfer recommendations. Therefore, users can set a threshold θ to determine the optimal order k, where k is set as the smallest value that satisfies

$$H(g|S_{(k)}) - H(g|S_{(k+1)}) < \theta. \tag{4}$$

3.2. Routing Recommendation

Our resolution network captures the likelihood that a ticket would be transferred to a group, given the past group transfer information. Then it makes ticket routing recommendations based on the majority of the decisions made in the past. Note that the right resolver group for a ticket is unknown at the beginning of ticket routing. What we know is the initial group that a problem ticket was assigned by the help desk according to the reported problem symptom.

To make effective ticket routing recommendations, we use the resolution network to guide the ticket routing. Starting from a first order model, we can make a decision solely based on the current group, at each step, choosing the next node with the highest transition probability $H(g|S_{(k)})$. Sometimes incorrect "local" decisions may cause unsuccessful transfers. To overcome this problem, we choose the next group based on the transition probabilities from one of the past states (called multiple active state), i.e., a subset of $S_{(k)}$.

We introduce a heuristic search algorithm, *Variable-order Multiple active state Search* (VMS). Here, we assume a ticket should not visit the same group twice. VMS considers a variable order of Markov models simultaneously (i.e. a variable length of past transfer patterns are considered), as well as multiple active states to advance the routing sequence.

In the following sections, we will use the VMS algorithm as a virtual ticket routing engine to simulate the resolution network. Note that VMS is not necessarily a perfect routing recommendation system, as both correct and wrong routing decisions are modeled in VMS and subsequently are used in routing decision making. However, it indeed provides a simulation of the real-life resolution network, which is valuable to our exclusive framework for assessing expertise awareness. We can easily knock out a node g_i in VMS by modifying its transfer probability $P(g_i|S_{(k)})$ as 0 for all $S_{(k)}$. It is beneficial to improve the quality of VMS by taking the ticket content into consideration, which is currently in development. Certainly, a high-quality automatic routing engine will make the analysis result more reliable.

4. Transferrer Role Analysis

In this section, we discuss how to evaluate "transfer effectiveness" of an expert group in a resolution network. In order to assess the transfer effectiveness of an expert group g_x, we compare the performance of ticket routing in the presence of g_x with the routing performance in the absence of g_x. The performance difference implies the transfer effectiveness of g_x.

The **Exclude** algorithm is presented in algorithm 1. First, we construct a resolution social network M using all routing sequences in the historical tickets, according to the technique discussed in Section 3.1. Then we simulate the effect of removing group g_x in the resolution network by constructing a virtual social network M' using routing sequences excluding g_x. Note that we focus on the analysis of transfer effectiveness of a group, not the technical expertise. Hence we only consider the sequences where g_x is not the resolution group. At last, we compare the MSTR difference of M and M'. The detail steps are described as below.

4.1. Virtual Resolution Network Construction

To simulate the resolution network in the absence of an expert group g_x, we construct model M' using all available information in the historical routing sequences except the one related g_x. For each ticket resolution sequence $G_{(k)}$ that contains g_x, we separate it into two segments: the first segment is from the beginning of $G_{(k)}$ to the group right before g_x, and the second segment is from the group right after to g_x to the end of the sequence. For example, given a ticket sequence $G_{(k)} = A \rightarrow B \rightarrow X \rightarrow C \rightarrow D$, when considering X's separation, we break it to $G_{(k_p)} = A \rightarrow B$ and $G_{(k_f)} = C \rightarrow D$. The original sequence $G_{(k)}$ is then replaced with $G_{(k_p)}$ and $G_{(k_f)}$. By doing so, we eliminate the influence of X.

After processing all the routing sequences that involve g_x, we build up a virtual model M' with the method proposed in Section 3.1, which represents the resolution network in the absence of group g_x.

Algorithm 1 Exclude g_x

Construct M with all the sequences ;
for resolution sequence $G_{(k)}$ **do**
    Separate $G_{(k)}$ into $G_{(k_p)}$ and $G_{(k_f)}$;
    Replace $G_{(k)}$ with $G_{(k_p)}$ and $G_{(k_f)}$;
end for
Construct M' by $G_{(k)}$ (Section 3.1);
for problem j which contains group g_x and starts with g_s **do**
    if $g_x \neq g_s$ and g_x is not the resolver **then**
        step($M'(g_s)$) = VMS(g_s, M');
        step(M(g_s)) = VMS(g_s, M);
        calculate improvement ratio Imp(j) using Eq.7;
    end if
end for
Significant Test for all (step($M(g_s)$),step($M'(g_s)$)) in participate sets PS(g_x, M) and PS(g_x, M') ;
if M and M' are significantly different **then**
    Ranking(Imp(g_s));
end if

4.2. Significance Test

To determine whether model M' (in absence of an expert group g_x) has significant difference with model M (in the presence of g_x) , we adopt standard T-test.

Only the routing sequences that have expert group g_x involved would have a different routing sequence in the absence of g_x, and only those sequences can provide useful information to evaluate the transfer effectiveness of g_x. Hence, to evaluate g_x, we only consider the resolution sequences that contain g_x when apply model M, referred as *participate set*, $PS(g_x, M)$ = $\{G_j\}(j = 1, ..., n)$. We use VMS algorithm to simulate the mostly likely routing decisions, to get $PS(g_x, M)$. Similarly, for virtual model M', we can get its participate set $PS(g_x, M')$ which is the set of resolution sequences contain g_x. For each problem ticket in $PS(g_x, M)$ with initial group g_s, we apply VMS to get the number of routing steps, represented as $step(M(g_s))$, and the number of routing steps using model M', as $step(M'(g_s))$. Considering all the instances with $step(M(g_s))$ and $step(M'(g_s))$ in $PS(g_x, M)$ and $PS(g_x, M')$, we compare the difference between these two models. We use X to represent the distribution of model M on $PS(g_x, M)$, where $X \sim N(\mu_1, \sigma_1^2)$, and Y to represent the distribution of M' on $PS(g_x, M')$ where $Y \sim N(\mu_2, \sigma_2^2)$, and $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$ are unknown variables. We first make a hypothesis: the means of two normally distributed populations are equal, which is $H_0 : \mu_1 = \mu_2$. $PS(g_x, M)$

and $PS(g_x, M')$ can be treated as a sample from the whole space, and the number of independent instances of X is m which is equal to the size of $PS(g_x, M)$, and the number of independent instances of Y is n which is equal to the size of $PS(g_x, M')$. We record their means and variances as following:

$$\overline{X} = \frac{1}{m}\sum_{i=1}^{m} X_i, \quad \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i,$$

$$S_{1m}^2 = \frac{1}{m}\sum_{i=1}^{m}(X_i - \overline{X})^2, \quad S_{2n}^2 = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \overline{Y})^2.$$

The t statistic can be calculated as follows:

$$t = \frac{\overline{X} - \overline{Y}}{\sqrt{mS_{1m}^2 + nS_{2n}^2}}\sqrt{\frac{mn(m+n-2)}{m+n}}, \quad (5)$$

where s_{1m}^2 and s_{2n}^2 are unbiased estimators of the variances of two sets. Compare the calculated t-value with the threshold chosen for statistical significance α (usually α is 0.10, 0.05, or 0.01 level). If $|t| \le t_{\frac{\alpha}{2}}$, then the hypothesis that the two models do not differ is accepted. Intuitively, if the change is not significant, then the presence or the absence of g_x has limited influence to the resolution network M, and thus has low significant impact to problem solving. Otherwise ($|t| > t_{\frac{\alpha}{2}}$), the hypothesis that two models do not differ is rejected. That is to say, the result is in favor of an alternative hypothesis, which typically states that two models do differ, and thus g_x is likely to play an important social role in the resolution network and in problem solving.

4.3. Positive or Negative Influence

After we differentiate groups with significant impact from the ones with insignificant ones using hypothesis test, the next question is how we can determine whether the impact is positive or negative.

Given an instance j in $PS(g_x, M')$, the ratio of the routing steps required for j using model M over the ones using model M', represented as Imp_j, is calculated as following:

$$Imp_j = \frac{step(M(g_s)) - step(M'(g_s))}{step(M(g_s))}. \quad (6)$$

The average changes to the routing steps to the whole participate set is

$$Imp(g_x) = \frac{\sum_{j=1}^{m} Imp_j}{|PS(g_x, M')|}. \quad (7)$$

g_x	MSTR(M)	MSTR(M')	Imp(g_x)
AUSFS	4.18	2.65	15.1%
UDCDS	4.16	2.846	7.89%
UDBTVAIXW	3.9	2.9076	7.45%
AIXCRM	3.27	2.465	6.45%
AUSAIX	3.43	2.333	3.94%
DSMOP	4.14	2	3.1%

Table 2. Role Effectiveness Results

If an expert group g_x has a positive $Imp(g_x)$ value, it indicates that the routing sequences are shortened in the absence of g_x. This indicates a negative impact of g_x in ticket routing. On the other hand, if an expert group g_x has a negative $Imp(g_x)$, then g_x plays a positive role in the ticket transfer. The higher absolute value of $Imp(g_x)$, the larger impact g_x has.

Our **Exclude** algorithm can provide a better understanding of the transferrer role of expert groups. For the groups that play an ineffective transferrer role and have limited contributions to the resolution network, it is important to target training programs to these groups to improve their expertise awareness of other groups.

5. Experiments

In this section, we report empirical studies of our approach on problem tickets collected from IBM's problem management system over a 1-year period from Jan 1, 2006 to Dec 31, 2006. These tickets were classified into 553 problem categories [1], e.g., `AIX`, `DB2`, `Windows`, etc. On average, $50 - 1,000$ groups were involved in solving tickets of each problem category.

In our study, we partition the data into training and testing sets for each problem category. Using the training set, we build a ticket routing engine presented in Section 3. After that, we analyze the transferrer roles on the testing dataset. We randomly choose the AIX problem category for case study.

For all the expert groups in `AIX` problem category, we first use the method described in Section 4 to do significance tests. For the groups whose absence causes significant changes to the model, we further investigate their impact to the ticket routing efficiency measured by the improvement ratio. Table 2 presents the top 6 groups with highest improvement ratios. Since the removal of such groups can result in shorter routing sequences, these are the groups whose ticket routing skills need to be trained.

As we can see, the expert group with the biggest improvement ratio is group `AUSFS`. To better under-

1. When a ticket is first opened, the helpdesk assigns a parameter that indicates which problem domain it falls into.

stand the result and validate our method, we check the tickets that `AUSFS` involves. We find that there are multiple similar tickets about disk operation errors handled by `AUSFS`. In Table 3, we show a sample ticket that involves `AUSFS`: after `AUSFS` received the ticket, it did not provide any useful problem diagnosis, but simply passed the unsolved ticket to other groups seeking for solution. It implies that `AUSFS` didn't positively contribute to problem solving of this ticket, and its absence in the model may shorten the routing sequences. This case study shows that the groups evaluated with an ineffective transferrer role in the resolution network should receive awareness training.

6. Related Work

Social network analysis (SNA) is the study of mathematical models for interactions among people, organizations and groups. Historically, research in the field has been led by social scientists and physicists [7], [17], [1], [19], [11]. With the popularity of Web 2.0 websites, social network analysis has attracted a lot of computer scientists' attention.

Social Role Analysis The use of social networks to discover "roles" for the people (or nodes) in the network goes back over three decades to the work of Lorrain and White [7]. It is based on the hypothesis that nodes on a network that relate to other nodes in "equivalent" ways must have the same role. Social network analysis is also introduced to identify important components or properties of network systems, such as the most striking one, the small world phenomenon [18] and the power-law degree distribution [3]. Analyzing network properties is also used for identifying authorities and search algorithms [2], [4], and discovering the network value of customers [12]. In element level analysis, several different centrality metrics such as degree, betweeness, clustering coefficient, and PageRank metrics have been proposed as graph theoretic metrics are capable of identifying the nodes' roles [15], e.g., finding particular nodes with an inordinately high number of connections, e.g., [7], [19]. Using these properties, roles are assigned to nodes, and role analysis becomes a well-studied topic in the social network research [13], [11]. However, it is clear that network properties are not enough to discover an expert's role effectiveness in a social network.

In this paper, instead of performing structural analysis, we analyze role effectiveness of expert groups from a functional perspective in the context of problem resolution.

Social Relationship The evaluation of social relationships mostly relies on sociologists [8] as well as computer scientists from the research area of human computer interaction. A definition that catches the essence of awareness in a broad way is the one suggested by Dourish et al. [6] where awareness is defined as "the understanding of the activity of the others, which provides a context of your own activity". Kraut et al. [9] show that geographical proximity is fundamental for the development of personal relations and communication. This includes first of all the knowledge of persons availability, both physical and emotional. Another aspect is that understanding how members' knowledge is used in a group. Some studies [10] claim that groups tend to be organized in knowledge networks where people relate to the knowledge of others. Hence, providing information about that knowledge is important, as it will increase the potential of collaboration within a group. [16] describes a system that provides computer support to bridge the gaps between people. The system can enhance the awareness and consciousness among lab members.

In this paper, we develop a novel framework to analyze the interaction of expert groups, and evaluate the effectiveness of their communications in a resolution social network.

7. CONCLUSIONS

In this paper, we propose an innovative framework to evaluate expertise awareness among different groups in a resolution social network in the context of problem solving. We explore an exclusive model to measure the knock-out values of expert groups. The proposed framework is evaluated on a large set of real-world problem tickets, which validate that our algorithm can successfully capture the weakest components in a resolution network. In future, we will explore how to analyze the relationship among experts using the exclusive model, extending our preliminary study [5]. We are also interested in studying how to optimize the automatic ticket routing engine to further improve the quality of the patterns discovered in this study.

Acknowledgment

This material is based on work partially supported by NSF IIS-0915438 and IIS-0917228.

References

[1] Reka Zsuzsanna Albert. *Statistical mechanics of complex networks*. PhD thesis, Notre Dame, IN, USA, 2001. Director-Albert-Laszlo Barabasi.

Entry	Description
New Ticket 29509068	disk operation error: This critical AIX_HD_ERROR was received by TEC server
Transferred to Group DSMOP	transfer to AUSFS for support
Transferred to Group AUSFS	RTPDSMOP transferred ticket to support
Transferred to Group WEDDELL	Resolution: pdisk has not been rejected from RAID, closing PR.

Table 3. Sample Ticket (Ineffective Transferrer: AUSFS)

[2] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Trans. Internet Technol.*, 5(1):231–297, 2005.

[3] Tian Bu and Donald F. Towsley. On distinguishing between internet power law topology generators. In *INFOCOM*, 2002.

[4] Yichuan Cai and Yi Chen. Mass: a multi-facet domain-specific influential blogger mining system. In *ICDE*, pages 1109–1112, 2010.

[5] Yi Chen, Shu Tao, Xifeng Yan, Nikos Anerousis, and Qihong Shao. Assessing expertise awareness in resolution networks. Technical Report TR-10-005, Arizona State University, 2010. http://wise.asu.edu/papers/tr10005.pdf.

[6] Paul Dourish and Victoria Bellotti. Awareness and coordination in shared workspaces. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114, New York, NY, USA, 1992. ACM.

[7] F.Lorrain and H.C. White. Structural equivalence of individuals in social networks. *Journal of Mathematical Sociology*, 1, 1971.

[8] Misook Heo. Relationship-based social awareness disclosure. In *ICALT '06: Proceedings of the Sixth IEEE International Conference on Advanced Learning Technologies*, pages 852–853, Washington, DC, USA, 2006. IEEE Computer Society.

[9] Robert Kraut, Carmen Egido, and Jolene Galegher. Patterns of contact and communication in scientific research collaboration. In *CSCW '88: Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, pages 1–12, New York, NY, USA, 1988. ACM.

[10] Hans Marmolin, Yngve Sundblad, and Björn Pehrson. An analysis of design and collaboration in a distributed environment. In *ECSCW'91: Proceedings of the second conference on European Conference on Computer-Supported Cooperative Work*, pages 147–162, Norwell, MA, USA, 1991. Kluwer Academic Publishers.

[11] Andrew Mccallum, Andr Corrada-emmanuel, and Xuerui Wang. Topic and role discovery in social networks. In *IJCAI*, pages 786–791, 2005.

[12] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. pages 61–70. ACM Press, 2002.

[13] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. Node roles and community structure in networks. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 26–35, New York, NY, USA, 2007. ACM.

[14] Qihong Shao, Yi Chen, Shu Tao, Xifeng Yan, and Nikos Anerousis. Efficient ticket routing by resolution sequence mining. In *Proc. ACM SIGKDD*, 2008.

[15] S.Wasserman and K.Faust. Social network analysis: Methods and applications. *Cambridge University*, 1994.

[16] Konrad Tollmar, Ovidiu Sandor, and Anna Schömer. Supporting social awareness @ work design and experience. In *CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 298–307, New York, NY, USA, 1996. ACM.

[17] Stanley Wasserman, Katherine Faust, and Dawn Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.

[18] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[19] Duncan J. Watts. *Six Degrees: The Science of a Connected Age*. W. W. Norton, New York, 2003.