Electronic Preprint for Journal of Information Processing Vol.20 No.3

Regular Paper

Estimating Message Importance Using Inferred Inter-recipient Trust for Supporting Email Triage

Sho Tsugawa[1,a)] Hiroyuki Ohsaki[1] Makoto Imase[1]

Received: August 31, 2011, Accepted: March 2, 2012

Abstract: In this paper, we propose a method to prioritize email using a trust network among users for supporting email triage, and evaluate its effectiveness with extensive experiments. In recent years, the amount of email received by individuals has increased, and therefore the time required for email triage (i.e., the process of going through unhandled email messages and deciding what to do with them) has therefore been increasing. Golbeck et al. proposed TrustMail, a prototype email client that prioritizes email in user's mailbox using a trust network (i.e., a social network representing trust relationships among users). In this paper, we extend the TrustMail concept to allow message-based email prioritization using inter-recipient trust, which is inferred trust score from the recipient to other recipients. We propose a method called *EMIRT* (*Estimating Message Importance from inter-Recipient Trust*) for enabling message-based prioritization. Through extensive experiments utilizing two email datasets, we quantitatively evaluate the effectiveness of EMIRT for email prioritization. Our experimental results show that EMIRT is effective for email prioritization. Specifically, our results show that EMIRT achieves significantly higher recall and precision than TrustMail in both email datasets and that EMIRT hardly gives low scores to urgently replied email (i.e., EMIRT achieves a very low false negative).

Keywords: trust network, email triage, email prioritization, social network

1. Introduction

In recent years, the amount of email received by individuals has increased, and therefore the time required for *email triage* (i.e., the process of going through unhandled email messages and deciding what to do with them) has also increased [1], [2]. For instance, 16% of employees in a corporation were found to have spent one hour or more per day just for email triage [1]. Moreover, among heavy email users receiving more than 100 messages per day, 46% spent one or more hour per day just for email triage [1].

At the same time, the use of trust information is currently popular in social networking services. For instance, the social networking service Orkut [3] allows participants to assign one of four levels of trust score to their acquaintances, and the assigned trust levels are visible to other users. In other services such as Moleskiing [4] and FilmTrust [5], participants are allowed to give trust scores to their acquaintances. Likewise, the consumer review site Epinions.com [6] permits users to give trust scores to other users.

Golbeck et al. [7] proposed TrustMail, a prototype email client that prioritizes email in user's mailbox using a trust network. The trust network is expressed by a directed graph whose edges are weighted by trust scores. TrustMail is a pioneering work in its use of trust networks to email triage. TrustMail assumes that trust scores on his/her acquaintances have been entered by a person and the trust network is accessible.

TrustMail has several clear advantages. In particular, TrustMail can prioritize email from unknown senders. When a user receives email from an unknown sender, TrustMail prioritize the email by inferring a trust score from the recipient to the sender (i.e., sender trust) by assuming the *transitivity* of trust relationships among users (e.g., if A trusts B and B trusts C, then A should trust C) [7]. Moreover, the burden on users to configure TrustMail is minimal, as users are only required to input and update the trust scores to their acquaintances in the trust network, which is information that is unlikely to change frequently.

We believe the approach of TrustMail is novel, yet several open issues remain. The primary issue is that TrustMail uses sender-based prioritization. TrustMail prioritizes email only using sender trust. Thus, TrustMail gives the same priority to all email from the same sender, which could degrade the accuracy of email prioritization. In addition, TrustMail assumes that the trust network is readily accessible, whereas in reality, the trust network may be difficult to obtain. For TrustMail to prioritize email effectively, almost all email users would need to join a social networking service, and the trust scores with their acquaintances in this service would need to be entered and made publicly available. However, in actually, many email users do not join social networking services, and even if they do join, they may not give trust scores.

[1] Graduate School of Information Science and Technology, Osaka University, Suita, Osaka 565–0871, Japan
[a)] s-tugawa@ist.osaka-u.ac.jp

In this paper, we therefore extend the TrustMail concept to allow message-based email prioritization. In many cases, email has one sender and multiple recipients. Our key idea is prioritizing email by using not only the sender trust but also the inter-recipient trust, which is inferred trust score from the recipient to other recipients. We propose a method called *EMIRT* (*Estimating Message Importance from inter-Recipient Trust*). In addition, we propose a method for constructing an implicit trust network, which can be obtained easily from email logs in MTAs (Message Transfer Agents), rather than relying on users to explicitly enter information into a trust network. An implicit trust network can be constructed using the frequency of email exchanges and replies among email users. EMIRT can then substitute an implicit trust network for an explicit one.

We also quantitatively evaluate the effectiveness of EMIRT and TrustMail for email prioritization through extensive experiments using two email datasets (i.e., the Enron Email Dataset, a large email corpus [8], and email between graduate students in our laboratory). To the best of our knowledge, performance evaluation of trust-based email prioritization has not been performed. Although a prototype of TrustMail has already been implemented and preliminary investigations on the possibility of calculating sender trust scores have been performed [7], the effectiveness of TrustMail for email prioritization has not yet been fully explored. Intuitively, one would expected that trust-based email prioritization is helpful for email triage. However, the effectiveness of trust-based email prioritization for email triage needs to be evaluated quantitatively.

The remainder of this paper is organized as follows. In Section 2, we introduce related works concerning email triage, as well as studies using trust information for other purposes. The proposed method for message-based email prioritization using inter-recipient trust, EMIRT, is presented in Section 3. Section 4 outlines our extensive experiments conducted on the two email datasets to evaluate the effectiveness of EMIRT. Finally, Section 5 concludes this paper and discusses future works.

2. Related Works

In the literature, several approaches have been proposed to facilitate email triage [9], [10], [11], [12], [13] and these methods can be classified into three categories: *abstraction*, *classification*, and *prioritization*.

Email abstraction summarizes a large amount of email to help users quickly grasp the content of messages. For instance, Smaranda et al. have proposed an approach for creating a summary of many email messages by extracting keywords from the email content using machine learning [12].

Email classification categorizes a large amount of email based on predefined rules. For instance, Balter et al. have proposed a simple mechanism to automatically classify unread email in the user's mailbox into different folders based on predefined rules such as the number of recipients and the words contained in the subject [9]. Neustaedter et al. have proposed a user interface that enables users to dynamically sort email based on several statistical metrics such as the total number of email exchanges with the sender and the percentage of prior messages from the sender to which the user replied [11].

Email prioritization estimates the priority of each email. For instance, Dredze et al. have proposed a history-based approach for predicting whether an email requires a reply by using the email exchange history between the user and the sender [10]. Yoo et al. have proposed a machine-learning approach that prioritizes email using a support vector machine (SVM) model with supervised learning [13].

We believe that a promising approach for email prioritization is to utilize social networks among email users. Garriss et al. have proposed a simple binary (i.e., high and low) email prioritization application called ReliableEmail that automatically constructs a sender white list from a social network [14]. ReliableEmail traverses the social network and adds contacts to the sender white list based on the observed relationships (e.g., friends and friend-of-friend are added to the white list). Golbeck et al. have proposed an email prioritization system called TrustMail, which uses a trust network [7].

We should note that use of trust information and trust networks for recommendations has been actively studied in the literature [4], [5], [15], [16], [17], [18], [19], [20]. This prior research on item recommendation is similar to our research on email prioritization in that the goal is to infer the usefulness of information from trust scores. However, item recommendation and email prioritization are different in the type of information they target. While item recommendation methods target public information, email prioritization focuses private information. Since each item in an item recommendation is public to others, recommendations can be given by using collaborative filtering, which is a conventional technique for making recommendations [21]. On the contrary, collaborative filtering cannot be used for email prioritization since each message is private information.

3. EMIRT (Estimating Message Importance from inter-Recipient Trust)

3.1 Overview

In this section, we propose a method called EMIRT to enable message-based prioritization. While TrustMail prioritizes email by using sender trust, the proposed EMIRT prioritizes email by using not only sender trust but also the inter-recipient trust. We use the same algorithm as TrustMail for calculating inferred trust scores from the recipient to the sender and other recipients. In this section, we first explain the concept of the proposed method. Then, the trust inference algorithm used in TrustMail and EMIRT are explained briefly in Section 3.2. The detailed algorithm of EMIRT is explained in Section 3.3. Section 3.4 discusses the proposed method to create implicit trust network.

EMIRT prioritizes email based on the idea that a high sender trust and *a high inter-recipient trust* imply that an email has great importance. Email is widely used for multicast-style communication. Hence, in many cases, we can use not only the sender trust but also the inter-recipient trust. By taking advantage of multiple inferred trust scores, we expect that EMIRT can successfully implement message-based email prioritization.

Figure 1 shows four examples of email that can be received by Bob. Let us assume that the inferred trust scores from Bob to

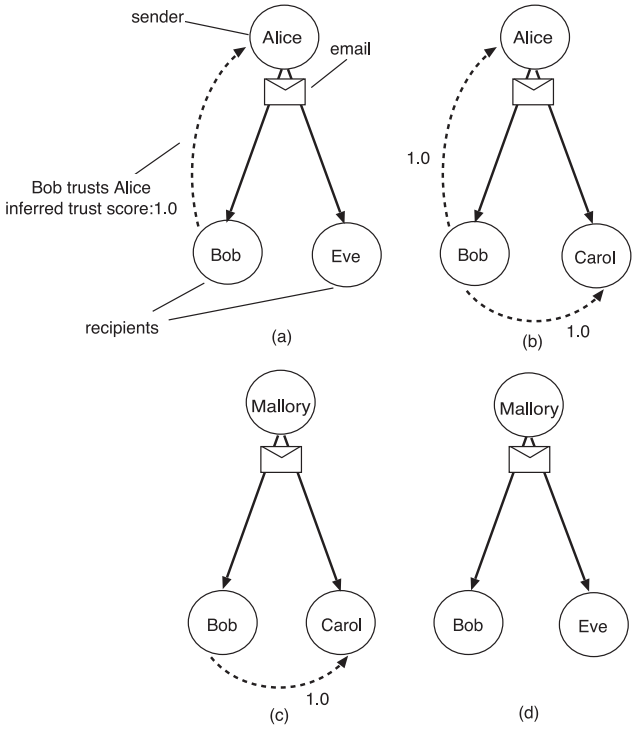

Fig. 1 Four examples of email reception: The inferred trust scores from Bob to Alice and Carol are 1, and those from Bob to Mallory and Eve are 0.

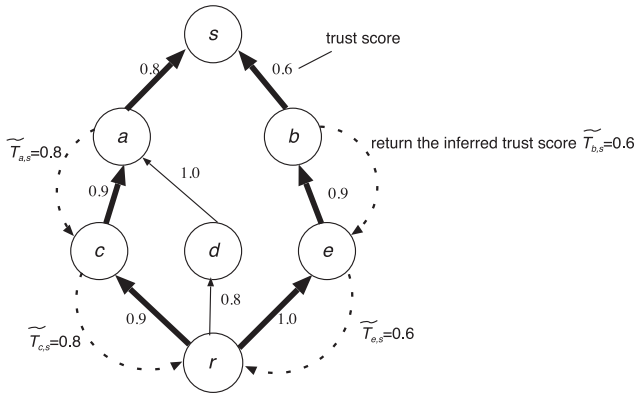

Fig. 2 An example of trust inference using a trust network: First, shortest paths from node r to the neighbor node of node s (i.e., node k's with $(k, s) \in E$) are obtained. There are three shortest paths: P_1: $r \to c \to a$, P_2: $r \to d \to a$, and P_3: $r \to e \to b$. Next, the minimum edge weight along each shortest path (i.e., path weight) is obtained. The path weights of path P_1, P_2, and P_3 are 0.9, 0.8, and 0.9, respectively. Then, only the paths with maximum path weight are utilized for calculating inferred trust score. In this case, path P_1 and P_3 are utilized. Utilizing path P_1, $\widetilde{T_{c,s}}$ is obtained by traversing the trust network. Node a returns its trust score of node s, $T_{a,s}$ (=0.8) as $\widetilde{T_{a,s}}$ to node c. Then, $\widetilde{T_{c,s}}$ is set to the equal value with $\widetilde{T_{a,s}}$ (=0.8). Similarly, utilizing path P_3, $\widetilde{T_{e,s}}$ is also obtained. Finally, $\widetilde{T_{r,s}}$ is obtained by calculating weighted mean of $\widetilde{T_{c,s}}$ and $\widetilde{T_{e,s}}$ weighted by $T_{r,c}$ and $T_{r,e}$. In this case $\widetilde{T_{r,s}}$ is $(0.8 \times 0.9 + 0.6 \times 1.0)/(0.9 + 1.0) \simeq 0.7$.

Alice and Carol are 1, and those from Bob to Mallory and Eve are 0. In cases (a) and (b), Bob receives an email message from Alice. Since the sender is the same, TrustMail gives equal priority to the email messages in (a) and (b). On the contrary, our EMIRT gives different priority scores to these email messages. Specifically, EMIRT gives a higher score to the message in (b), since Bob trusts both the sender (Alice) and the other recipient (Carol). However, the inter-recipient trust is different in situation (a), and the priority score of the message under EMIRT is lower since Bob does not trust the other recipient, Eve. In cases (c) and (d), Bob receives an email message from Mallory. Since the inferred trust score from Bob to Mallory is 0, TrustMail can not discriminate between the two messages. In contrast, EMIRT is able to prioritize these email, and a higher score is given to the message in (c) than in (d) since information on inter-recipient trust is available (i.e., Bob trusts Carol).

3.2 Trust Inference Algorithm

In this section, we briefly explain the algorithm for calculating an inferred trust score [7], [22], [23]. **Figure 2** illustrates an example of trust inference using a trust network. Refer to Refs. [7], [22], [23] for the details of the trust inference algorithm. Let a weighted directed graph $G = (V, E)$ be a trust network, where the weight of an edge $(i, j) \in E$ represents the trust score from i to j, which is denoted by $T_{i,j}$. We assume that $0 < T_{i,j} \leq 1$ and $T_{i,i} = 1$. If no edge exists from node i to node j, then the trust score from i to j is unknown.

Golbeck et al. proposed several trust inference algorithms for use with binary trust networks [7] and weighted trust networks [22], [23]. Since we use a weighted trust network in this paper, we introduce trust inference algorithm for weighted trust networks [22], [23].

Basically, an inferred trust score from node i to node s, which is denoted by $\widetilde{T_{i,s}}$, is calculated by recursively traversing the trust network using a breadth-first search (BFS) algorithm, as described below. This algorithm returns $\widetilde{T_{i,s}}$, as well as the path weight from node i to node s, denoted by $w_{i,s}$, which represents the minimum of link weights (trust scores) along the shortest path from node i to node k with $(k, s) \in E$, and the number of hops of the shortest path from node i to node s, denoted by $d_{i,s}$. Both of $w_{i,s}$ and $d_{i,s}$ are used in calculating an inferred trust score. Note that in the BFS algorithm, each node is visited at most once. When we calculate an inferred trust score from node r to node s, we make the originating node r be the current node i; $i \leftarrow r$, and the following algorithm is performed.

(1) Check acquaintances
If node i has a trust score to node s, return the trust score $T_{i,s}$. Namely, $\widetilde{T_{i,s}} \leftarrow T_{i,s}$, $d_{i,s} \leftarrow 1$, and $w_{i,s} \leftarrow 1$, and then return $\widetilde{T_{i,s}}$, $d_{i,s}$, and $w_{i,s}$ if $(i, s) \in E$. If node i has no neighbor to visit, $\widetilde{T_{i,s}} \leftarrow 0$, $d_{i,s} \leftarrow \infty$, and $w_{i,s} \leftarrow 0$, and then return $\widetilde{T_{i,s}}$, $d_{i,s}$, and $w_{i,s}$. Otherwise, proceed to the next step.

(2) Obtain trust scores from all neighbors
Ask all neighbors of node i to return their trust scores for node s. Namely, for all node j's with $(i, j) \in E$, obtain $\widetilde{T_{j,s}}$'s, $d_{j,s}$'s, and $w_{j,s}$'s by recursively performing the algorithm from the step (1) with making the current node be j.

(3) Calculate average trust score
Calculate the weighted mean of trust scores obtained from all neighbors as follows. First, calculate the number of hops from node i to node s through node j and the path weight from node i to node s through node j. Namely, $d_{i,j,s} \leftarrow d_{j,s} + 1$ and $w_{i,j,s} \leftarrow \min(w_{j,s}, T_{i,j})$ for all j's. Then, $\widetilde{T_{i,s}}$ is obtained as

$$\widetilde{T_{i,s}} = \frac{\sum_{\{j|w_{i,j,s}=max,\, d_{i,j,s}=h\}} T_{i,j}\, \widetilde{T_{j,s}}}{\sum_{\{j|w_{i,j,s}=max,\, d_{i,j,s}=h\}} T_{i,j}}, \quad (1)$$

where h is the minimum of $d_{i,j,s}$ for all j's, and max is the maximum of $w_{i,j,s}$ for j's with $d_{i,j,s} = h$. Finally, $d_{i,s} \leftarrow h$ and $w_{i,s} \leftarrow max$, and return $\widetilde{T_{i,s}}$, $d_{i,s}$ and $w_{i,s}$.

Note that if node r gives a trust score to node s (i.e., $(r, s) \in E$), $\widetilde{T_{r,s}}$ is the trust score $T_{r,s}$. If node s is not reachable from node r (i.e., there exists no path from node r to node s), $\widetilde{T_{r,s}} = 0$.

3.3 EMIRT Email Prioritization Algorithm

Symbolic notation used throughout this paper is illustrated in **Fig. 3**. This figure shows the example of email m being sent from sender s to three recipients $\mathcal{R}^m = \{r_1, r_2, r_3\}$.

EMIRT prioritizes email m by using the inferred sender trust score and inferred inter-recipient trust scores. Other recipients are identified from the header of the email (e.g., To and Cc fields). Specifically, the priority score of email m received by recipient r is given by

$$p_r^m = \eta^m \widetilde{T_{r,s}} + \xi^m \sum_{u \in \mathcal{R}^m} \widetilde{T_{r,u}}, \quad (2)$$

where $\widetilde{T_{i,j}}$ is an inferred trust score from i to j, which is explained in Section 3.2. Additionally, η^m and ξ^m are parameters that control the weightings of the sender trust and inter-recipient trust. For instance, let us assume $\eta^m = \xi^m = 1/(1 + |\mathcal{R}^m|)$. In Fig. 1 (a), the EMIRT score of the email from Alice to Bob is 2/3 since the inferred trust score from Bob to Alice, from Bob to Eve, and from Bob to himself are 1, 0, and 1, respectively. In Fig. 1 (b), the EMIRT score of the email from Alice to Bob is 1 since the inferred trust scores from Bob to Alice, from Bob to Carol, and from Bob to himself are all 1.

While the desired settings of the weights, η^m and ξ^m, would ideally take into account factors such as the objective of email communication and the style of email usage, we consider two weighting options for simplicity. The first option assigns equal weights to sender trust and the inter-recipient trust between each recipient ($\eta^m = \xi^m = 1/(1 + |\mathcal{R}^m|)$). The second option assigns equal weights to sender trust and the sum of all inter-recipient trust ($\eta^m = 1/2$, $\xi^m = 1/2|\mathcal{R}^m|$). These weighting methods are compared experimentally in Section 4.

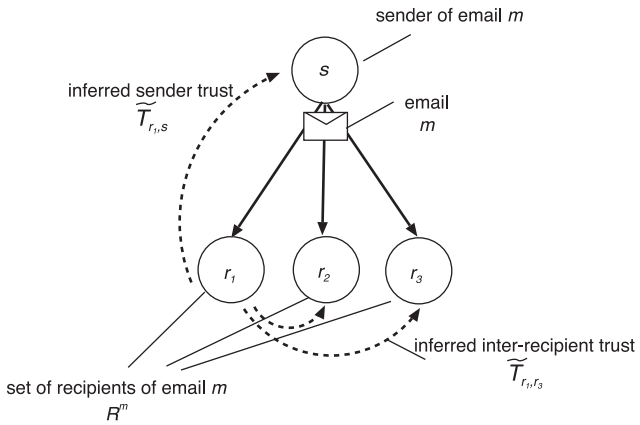

Fig. 3 Symbolic notation for email m being sent from sender s to three recipients $\mathcal{R}^=\{r_1, r_2, r_3\}$.

3.4 Implicit Trust Network Construction

In this section, we propose a method for constructing an implicit trust network using the email exchange history among email users. Such email history should be easier to obtain than an explicit trust network.

If user i sends email to user j frequently, we can assume that user i trusts user j. Moreover, if user i frequently replies to email from user j, we can assume that user i trusts user j. Based on these assumptions, the trust score from user i to user j is estimated as the linear combination of sending frequency from user i to user j and replying frequency from user i to user j. Namely, the trust score from user i to user j, $T_{i,j}$, is given by

$$T_{i,j} = w\,\lambda_{i,j} + (1 - w)\,\mu_{i,j}, \quad (3)$$

where $\lambda_{i,j}$ is the normalized sending frequency from user i to user j, $\mu_{i,j}$ is the normalized replying frequency from user i to user j, and w is a parameter. $\lambda_{i,j}$ is defined as

$$\lambda_{i,j} = 1 - \frac{1}{1 + \frac{N_{i,j}}{\widetilde{N_i}}}, \quad (4)$$

where $N_{i,j}$ is the number of email messages from user i to user j in a given observation period, and $\widetilde{N_i}$ is the median of $N_{i,j}$ for all j's. Similarly, $\mu_{i,j}$ is defined as

$$\mu_{i,j} = 1 - \frac{1}{1 + \frac{M_{i,j}}{\widetilde{M_i}}}, \quad (5)$$

where $M_{i,j}$ is the number of email messages replied from user i to user j in a given observation period, and $\widetilde{M_i}$ is the median of $M_{i,j}$ for all j's. $\lambda_{i,j}$ is 0 if i sends no email messages to j, is 0.5 if i sends $\widetilde{N_i}$ messages to j, and approaches towards 1 as $N_{i,j}$ increases. Similarly, $\mu_{i,j}$ is 0 if i sends no replies to j, is 0.5 if i sends $\widetilde{M_i}$ replies to j, and approaches towards 1 as $M_{i,j}$ increases. Hence, $T_{i,j}$ is 0 if i sends no email messages to j, and approaches towards 1 as $N_{i,j}$ and $M_{i,j}$ increase. $T_{i,j}$ is 0.5 if $N_{i,j} = \widetilde{N_i}$, $M_{i,j} = \widetilde{M_i}$, and $w = 0.5$.

4. Experiments

4.1 Experiments with Enron Email Dataset

4.1.1 Methodology

We evaluate the effectiveness of our proposed method for prioritizing email through experiments using the *Enron Email Dataset* [8], a large email corpus. The Enron Email Dataset contains 252,759 email messages with headers and body texts of 151 users in the Enron Corporation. To the best of our knowledge, the Enron Email Dataset is the only real corporate email dataset that is publicly available, and it has been used for several studies (for example, see Refs. [24], [25], [26] and the references therein). Because of its size and availability, Enron Email Dataset should be useful for evaluating the effectiveness of EMIRT in prioritizing email. For evaluation, we perform preprocessing to the dataset by removing duplicate email. Moreover, 13 inactive users who sent less than 21 email messages over the two-year period from April 1, 2000 to March 31, 2001 are removed in order to exclude users who have email accounts but rarely used them. The average and median number of email sent by the 151 users are 1,310 and 408, respectively.

To evaluate the effectiveness of EMIRT, we investigate the correlation between the estimated importance of an email calculated by the proposed method and the actual time-to-reply in the dataset. Namely, we examine the effectiveness of EMIRT in identifying urgent email. For comparison purposes, we also investigate the correlation between the inferred sender trust of email, which is equivalent to the score of TrustMail, and the time-to-reply. Note, in this experiment we can not evaluate the effectiveness of EMIRT in estimating the importance of unreplied to, but nevertheless important, email.

The time-to-reply for each email in the Enron Email Dataset is obtained as follows. Since the `Reply-To` fields are missing in the email headers in the Enron Email Dataset, we determine the original email and the replying email by using the subject line of the email. Specifically, if a user receives an email and he/she returns an email to the sender with the same subject line, but with a prefix `Re:`, those two email are considered as the original email and the replying email, respectively. The time-to-reply for the original email is obtained as the elapsed time between receiving the original email and sending the replying email. Note that not all replying email could be discovered since we simply identified replying email through their subject lines. The 25th, 50th, and 75th percentiles of the time-to-reply for all replied email are 0.28 hours, 1.6 hours, and 16.4 hours, respectively. Approximately 2% of all email in the dataset received a replying email.

In this experiment, we prioritize 105,677 email messages received during the period from April 1, 2001 to March 31, 2002 using EMIRT.

To prioritize each email, an implicit trust network is obtained using email received after April 1, 2000 and before the arrival of the email to be prioritized according to the method explained in Section 3.4. Unless explicitly stated, we use $w = 0.5$ as the weighting parameter for determining trust scores in the implicit trust network.

We used $\eta^m = \xi^m = 1/(1 + |\mathcal{R}^m|)$ to calculate EMIRT scores in the following experiments.

4.1.2 Correlation between EMIRT Score and Time-to-reply

We first investigate the relation between EMIRT score of an email and its time-to-reply. A boxplot that represents relation between EMIRT score and time-to-reply is shown in **Fig. 4**. For comparison purpose, a boxplot representing the relation between TrustMail score and time-to-reply is shown in **Fig. 5**. In these figures, email are classified into five sets based on their time-to-reply: *not-replied, slowly-replied, medially-replied, fastly-replied, and urgently-replied*. Replied email are split into four sets (i.e., slowly-replied, medially-replied, fastly-replied, and urgently-replied) based on the quartile ranges. All non-replied email are classified as not-replied. The box in the figure indicates the range of values from the first quartile (25%-tile) to the third quartile (75%-tile) of EMIRT scores. The line within the box indicates the median (50%-tile). The ends of whiskers of the box are the lowest datum within 1.5 IQR (Inter Quartile Range) of the first quartile and the highest datum within 1.5 IQR of the third quartile [27]. Outliers are shown as lines with a dot.

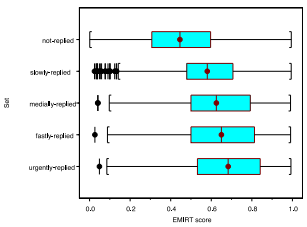

Fig. 4 Boxplot representing the relation between EMIRT score and time-to-reply (correlation coefficient $r = -0.077$).

Comparison of Figs. 4 and 5 indicates that both EMIRT and TrustMail scores have a weak, negative correlation with time-to-reply, and that EMIRT score has slightly stronger correlation than TrustMail score. The quartiles of EMIRT and TrustMail scores of email in urgently-replied sets are relatively higher than those in slowly-replied and not-replied sets. The correlation coefficient between EMIRT score and time-to-reply is -0.077, and the correlation coefficient between TrustMail score and time-to-reply is -0.072.

We then perform a more detailed analysis to investigate the relation between EMIRT/TrustMail score and time-to-reply. Histograms of EMIRT and TrustMail scores for the five sets are shown in **Fig. 6**.

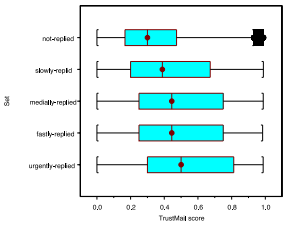

Fig. 5 Boxplot representing the relation between TrustMail score and time-to-reply (correlation coefficient $r = -0.072$).

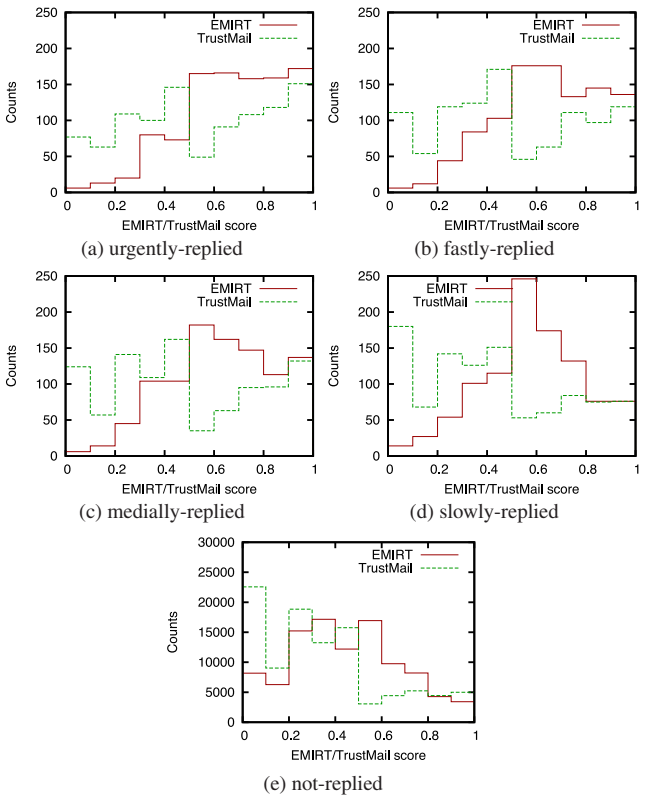

Fig. 6 Histograms of EMIRT and TrustMail scores for the five sets based on time-to-reply.

Table 1 Percentage of email that EMIRT and/or TrustMail cannot prioritize.

description	percentage	count
1) email from strangers (i.e., persons who are not directly connected with the recipients)	49%	52,229
2) email from strangers who are unreachable in the implicit trust network	21%	22,413
3) email from strangers, sent to recipients who are all strangers, all of whom are unreachable in implicit trust network	11%	11,944

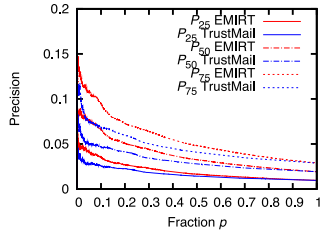

Fig. 7 Precision when the fraction p of email with high EMIRT/TrustMail scores is extracted.

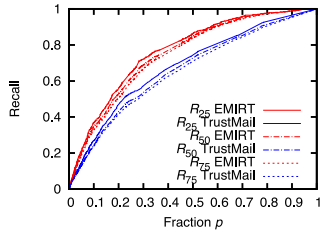

Fig. 8 Recall when the fraction p of email with high EMIRT/TrustMail scores is extracted.

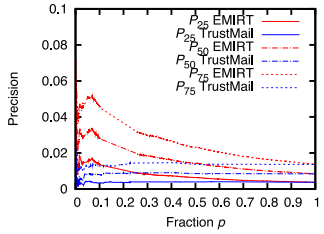

Fig. 9 Precision when the fraction p of email with high EMIRT/TrustMail scores is extracted (email from strangers only).

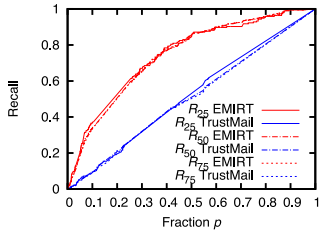

Fig. 10 Recall when the fraction p of email with high EMIRT/TrustMail scores is extracted (email from strangers only).

Figure 6 clearly shows that use of EMIRT results in more consistent prioritization than TrustMail. For instance, the histograms of EMIRT scores (Fig. 6 (a)–(d)) take an almost convex form, whereas the variability of TrustMail scores is high. Such a stable prioritization indicates that EMIRT rarely gives low scores to urgently replied email (i.e., EMIRT achieves a very low false negative). Since a low rate of false negative is required in email triage, these results suggest the effectiveness of the proposed EMIRT for email triage.

4.1.3 Performance Evaluation for Email Triage

We examine the effectiveness of EMIRT for email triage by looking at its *precision* and *recall*, which are common metrics to evaluate accuracy and completeness, respectively [28]. Namely, we investigate how EMIRT and TrustMail perform in these areas when identifying email requiring a fast reply. Precision and recall for retrieving the top N% quickly-replied email are denoted by P_N and R_N, respectively. Precision and recall, when the fraction p of email with high EMIRT/TrustMail scores is extracted, are shown in **Figs. 7** and **8**, respectively. Note that the precision with random extraction is equivalent to the value with $p = 1$.

First, we focus on precision (Fig. 7). Figure 7 shows that EMIRT achieves significantly higher precision than TrustMail. EMIRT achieves approximately 1.5 times more precision than TrustMail, and 3 to 5 times more precision than random extraction of messages. This observation suggests that EMIRT is helpful for identifying quickly replied email.

Second, we focus on recall (Fig. 8). Note that in general, a trade-off exists between precision and recall. Surprisingly, EMIRT achieves significantly higher recall than TrustMail does (Fig. 8). For instance, EMIRT achieves approximately 15% higher recall than TrustMail with $p = 0.5$. As we have discussed in Section 4.1.2, such a high recall implicitly suggests the effectiveness of EMIRT for supporting email triage.

4.1.4 Notes on Message-based Email Prioritization

Next, we investigate the effectiveness of the proposed EMIRT for message-based email prioritization. **Table 1** summarizes the fraction of email that EMIRT and/or TrustMail cannot prioritize, 1) the percentage of email from strangers (i.e., persons who are not directly connected with the recipients in the implicit trust network), 2) the percentage of email from strangers who are unreachable in the implicit trust network, and 3) the percentage of email from strangers, sent to recipients who are all strangers, all of whom are unreachable in the implicit trust network. Since TrustMail utilizes only the sender trust, TrustMail cannot prioritize email from strangers who are unreachable in the implicit trust network. Similarly, EMIRT cannot prioritize email whose senders and recipients are unreachable in the implicit trust network. Table 1 indicates that EMIRT fails to rate just 11% of email whereas TrustMail fails to prioritize 21% of messages.

To evaluate the effectiveness of EMIRT for message-based prioritization, we calculate the precision and recall only for email from strangers (**Figs. 9** and **10**). We select email from strangers, and calculate precision and recall for those email similarly to the previous section. Then, we examine the effectiveness of EMIRT for prioritizing email from strangers.

Again, these results (Figs. 9 and 10) show that EMIRT achieves significantly higher levels of precision and recall than TrustMail. Precision and recall with TrustMail are comparable with those

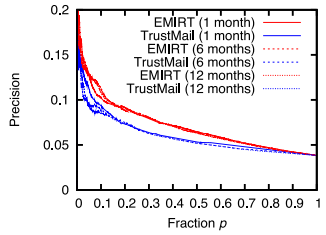

Fig. 11 P_{100} when the fraction p of email with high EMIRT/TrustMail scores is extracted over various time periods for creation of implicit trust network.

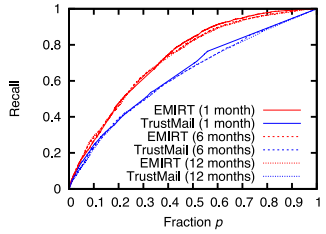

Fig. 12 R_{100} when the fraction p of email with high EMIRT/TrustMail scores is extracted over various time periods for creation of implicit trust network.

of random extraction, and little improvement can be seen over random extraction. On the contrary, EMIRT achieves significantly higher precision and recall than TrustMail. These observations suggest that EMIRT has higher accuracy since it conducts message-based prioritization, rather than sender-based prioritization.

4.1.5 Parameter Sensitivity Analysis

In this section, parameter sensitivity of our proposed EMIRT is examined.

First, we calculate precision and recall while changing the length of the time period over which information is taken from the dataset used to create the implicit trust network (**Figs. 11** and **12**). For each email message to be prioritized, implicit trust networks are created using email received during the one-month, six-month, and 12-month periods before the arrival of the email.

From these results (Figs. 11 and 12), one can see that precision and recall for the one-month dataset and 12-months dataset is not particularly different. When p is small, precision and recall for the 12-months dataset are slightly higher than for the one-month dataset. On the contrary, when p is large, precision and recall for the 12-months dataset is slightly lower than for the one-month dataset. For instance, when the fraction p is 0.1, recall for the 12-months dataset is approximately 0.29, whereas recall for the one-month dataset is approximately 0.27. Conversely, when p is 0.5, recall for the 12-months dataset is approximately 0.82 while recall for the one-month dataset is approximately 0.84. These results suggest that a long-term dataset is required for accurately extracting small numbers of replied email messages. However, these results also suggest that a month is long enough period to extract most replied email messages.

Second, we calculate precision and recall when changing parameter w, which is used in the creation of the implicit trust network (**Figs. 13** and **14**). From these results (Figs. 13 and 14), we can see that precision and recall with EMIRT do not drastically change when w is changed. However, when w is set to either 0 or 1, precision and recall are slightly lower than for other values

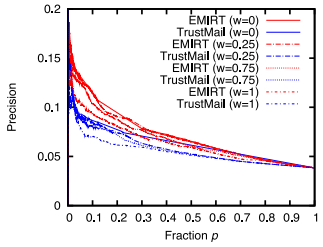

Fig. 13 P_{100} when the fraction p of email with high EMIRT/TrustMail scores is extracted for various values of parameter w used in creation of the implicit trust network.

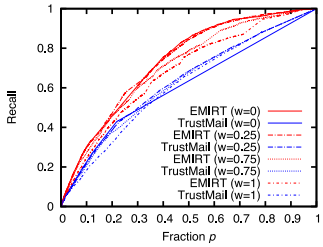

Fig. 14 R_{100} when the fraction p of email with high EMIRT/TrustMail scores is extracted for various values of parameter w used in creation of the implicit trust network.

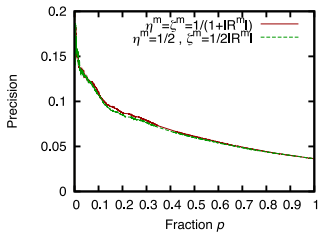

Fig. 15 P_{100} when the fraction p of email with high EMIRT scores is extracted using different values for parameter η^m and ξ^m to calculate EMIRT score.

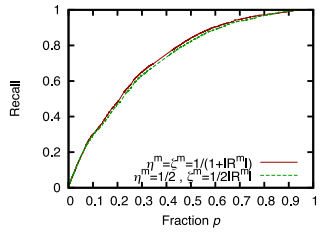

Fig. 16 R_{100} when the fraction p of email with high EMIRT scores is extracted using different values for parameters η^m and ξ^m to calculate EMIRT score.

of w's. Moreover, using a smaller value for w slightly improves precision and recall. These results suggest that the accuracy of EMIRT can be slightly improved by choosing an optimal value for w. However, drastic improvements are not to be expected.

Third, we calculate precision and recall under two settings of η^m and ξ^m, which are the parameters used in EMIRT score calculations (**Figs. 15** and **16**). Namely, we use (a) $\eta^m = \xi^m = 1/(1 + |\mathcal{R}^m|)$ and (b) $\eta^m = 1/2$, $\xi^m = 1/2|\mathcal{R}^m|$. From these results (Figs. 15 and 16), one can see that precision and recall with parameters $\eta^m = \xi^m = 1/(1 + |\mathcal{R}^m|)$ are slightly higher than those with parameters $\eta^m = 1/2$, $\xi^m = 1/2|\mathcal{R}^m|$. However, the differences are not so significant in practice, since the difference in recall is less than two percent.

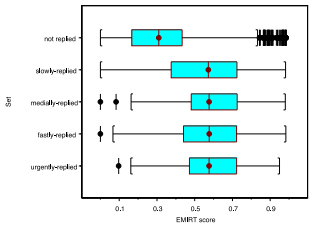

Fig. 17 Boxplot representing the relation between EMIRT score and time-to-reply (correlation coefficient $r = -0.28$).

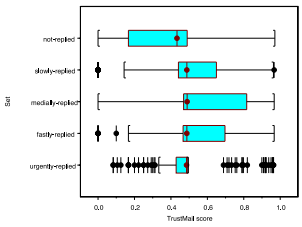

Fig. 18 Boxplot representing the relation between TrustMail score and time-to-reply (correlation coefficient $r = -0.21$).

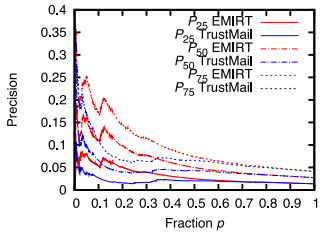

Fig. 19 Precision when the fraction p of email with high EMIRT/TrustMail scores is extracted.

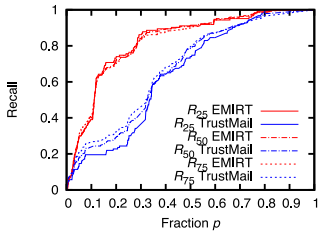

Fig. 20 Recall when the fraction p of email with high EMIRT/TrustMail scores is extracted.

4.2 Experiments with Laboratory Email Logs

4.2.1 Methodology

Next we evaluate the effectiveness of EMIRT in estimating importance of email through experiments using 15,070 email messages received by three graduate students in our laboratory during the one-year period from April 1, 2010 to March 31, 2011. Note that spam email is excluded by the spam filters used by the students.

We conduct similar experiments in Section 4.1 using the same parameter configurations. We prioritize 8,982 email messages that were received during the six-month period period from September 1, 2010 to March 31, 2011. For each email to be prioritized, an implicit trust network is created using the method explained in Section 3.4 with the email received after April 1, 2010 to before the arrival of the email to be prioritized. Note that the time-to-reply for each email is obtained using `In-Reply-To` field in the email header. The 25th, 50th, and 75th percentiles of the time-to-reply for all replied email are 0.4 hours, 2.0 hours, and 10.6 hours, respectively.

4.2.2 Correlation between EMIRT Score and Time-to-reply

We first investigate the relation between the EMIRT score of an email and its time-to-reply. Similarly to the experiments in Section 4.1, a boxplot representing the relation between EMIRT score and time-to-reply is shown in **Fig. 17**. For comparison purpose, a boxplot representing the relation between TrustMail score and time-to-reply is shown in **Fig. 18**.

These results (Figs. 17 and 18) show that both EMIRT and TrustMail scores have a weak, negative correlation with time-to-reply. Compared to the experiments with Enron Email Dataset, the correlation between EMIRT score and time-to-reply is stronger. The correlation coefficient between EMIRT score and time-to-reply is -0.28, and the correlation coefficient between TrustMail score and time-to-reply is -0.21.

4.2.3 Performance Evaluation for Email Triage

We examine the effectiveness of EMIRT for email triage by investigating its levels of precision and recall. Precision and recall, when the fraction p of email with high EMIRT/TrustMail scores is extracted, are shown in **Figs. 19** and **20**, respectively.

Similar to the experiments with Enron Email Dataset, these results (Figs. 19 and 20) show that EMIRT achieves significantly higher precision and recall than TrustMail. In contrast to the experiments with Enron Email Dataset, EMIRT achieves significantly higher recall than TrustMail, even when the fraction p of email messages extracted is small.

Although we cannot overly generalize from our experiments performed on only two datasets, our results suggest that EMIRT could be an effective method for email triage in several environments. Experimental results using email in our laboratory are similar to those with the Enron Email Dataset. Hence, we expect that EMIRT could be effective for email triage in a variety of environments.

5. Conclusion and Future Works

In this paper, we have proposed a method called EMIRT for enabling message-based email prioritization. EMIRT can achieve better email prioritization since it utilizes inter-recipient trust, in addition to sender trust.

Furthermore, through extensive experiments using two email datasets, the Enron Email Dataset [8] and email among graduate students in our laboratory, we have quantitatively evaluated the effectiveness of EMIRT for email prioritization. Our results show that EMIRT achieves high recall, which is favorable for supporting email triage. In particular, approximately 85% of quickly replied email messages are identified when 50% of email messages are extracted. On the contrary, our results show that the precision of EMIRT is not as high as we would hope. One reason for this low precision is that only a small proportion of all email normally receive a reply. However, improved precision is required to reduce the burden on users to triage email.

Improving the accuracy of email prioritization, using information in addition to trust networks could be promising. For instance, the context of an email is considered useful for prioritization. In Ref. [29], email with social context and emailwith information requests have been shown to prompt email replies. The

literals contained in an email would also be useful information to consider. In Ref. [10], the presence of question marks has been shown to affect replying actions. Moreover, the use of machine learning techniques such as SVMs are effective [13]. Since the time-to-reply can be obtained using the email exchange history, supervised learning can be performed without explicit input by users.

Acknowledgments The authors would like to thank Prof. Masayuki Murata for his kind support and valuable discussions.

This research was partly supported by "Global COE (Centers of Excellence) Program" of the Ministry of Education, Culture, Sports, Science and Technology, Japan and Grant-in-Aid for Scientific Research (B) (21300022).

References

[1] Neustaedter, C., Brush, A.B. and Smith, M.A.: Beyond "From" and "Received": Exploring the Dynamics of Email Triage, *Proc. ACM Conference on Human Factors in Computing Systems* (*CHI '05*), pp.1977–1980 (2005).
[2] Dabbish, L.A. and Kraut, R.E.: Email overload at work: An analysis of factors associated with email strain, *Proc. 20th Anniversary Conference on Computer Supported Cooperative Work* (*CSCW '06*), pp.431–440 (2006).
[3] Orkut, available from ⟨http://www.orkut.com/⟩.
[4] Avesani, P., Massa, P. and Tiella, R.: A trust-enhanced recommender system application: Moleskiing, *Proc. 2005 ACM Symposium on Applied Computing* (*SAC '05*), pp.1589–1593 (2005).
[5] Golbeck, J. and Hendler, J.: FilmTrust: Movie recommendations using trust in Web-based social networks, *Proc. 3rd IEEE Consumer Communications and Networking Conference* (*CCNC '06*), pp.282–286 (2006).
[6] Epinion.com, available from ⟨http://www0.epinions.com/⟩.
[7] Golbeck, J. and Hendler, J.: Inferring binary trust relationships in Web-based social networks, *ACM Trans. Internet Technology*, Vol.6, No.4, pp.497–529 (2006).
[8] Shetty, J. and Adibi, J.: The Enron email dataset database schema and brief statistical report, Technical Report, Information Sciences Institute, University of Southern California (2004).
[9] Balter, O. and Sidner, C.L.: Bifrost inbox organizer:giving users control over the inbox, *Proc. 2nd Nordic Conference on Human-computer Interaction* (*NordiCHI '02*), pp.19–23 (2004).
[10] Dredze, M., Brooks, T., Carroll, J., Magarick, J., Blitzer, J. and Pereira, F.: Intelligent email: Reply and attachment prediction, *Proc. 13th International Conference on Intelligent User Interfaces* (*IUI '08*), pp.321–324 (2008).
[11] Neustaedter, C., Brush, A.B., Smith, M.A. and Fisher, D.: The social network and relationship finder: Social sorting for email triage, *Proc. 2nd Conference on Email and Anti-Spam* (*CEAS '05*) (2005).
[12] Muresan, S., Tzoukermann, E. and Klavans, J.: Combining Linguistic and Machine Learning Techniques for Email Summarization, *Proc. 2001 Workshop on Computational Natural Language Learning*, Vol.7, pp.1–8 (2001).
[13] Yoo, S., Yang, Y., Lin, F. and Moon, I.-C.: Mining social networks for personalized email prioritization, *Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (*KDD '09*), pp.967–975 (2009).
[14] Garriss, S., Kaminsky, M., Freedman, M.J., Karp, B., Mazieres, D. and Yu, H.: RE: Reliable Email, *Proc. 3rd Symposium on Networked Systems Design and Implementation* (*NSDI '06*), pp.297–310 (2006).
[15] Massa, P. and Avesani, P.: Trust-aware Collaborative Filtering for Recommender Systems, *Proceedings of Federated International Conference On The Move to Meaningful Internet: CoopIS, DOA, ODBASE*, pp.492–508 (2004).
[16] Sarda, K., Gupta, P., Mukherjee, D., Padhy, S. and Saran, H.: A Distributed Trust-based Recommendation System on Social Networks, *Proc. 2nd IEEE Workshop on Hot Topics in Web Systems and Technologies* (*HotWeb '08*), pp.1–6 (2008).
[17] Montaner, M., Lopez, B. and de la Rosa, J.L.: Opinion-Based Filtering through Trust, *Proc. 6th International Workshop on Cooperative Information Agents VI*, pp.164–178 (2002).
[18] Kinateder, M. and Rothermel, K.: Architecture and Algorithms for a Distributed Reputation System, *Proc. 1st International Conference on Trust Management* (*iTrust '03*), pp.1–16 (2003).
[19] Chen, M. and Singh, J.P.: Computing and using reputations for internet ratings, *Proc. 3rd ACM Conference on Electronic Commerce* (*EC '01*), pp.154–162 (2001).
[20] DuBois, T., Golbeck, J., Kleint, J. and Srinivasan, A.: Improving recommendation accuracy by clustering social networks with trust, *Proc. ACM RecSys 2009 Workshop on Recommender Systems and the Social Web*, pp.1–8 (2009).
[21] Sarwar, B., Karypis, G., Konstan, J. and Reidl, J.: Item-based collaborative filtering recommendation algorithms, *Proc. 10th International Conference on World Wide Web* (*WWW '01*), pp.285–295 (2001).
[22] Golbeck, J.: Personalizing applications through integration of inferred trust values in semantic web-based social networks, *Proc. Semantic Network Analysis Workshop*, pp.15–28 (2005).
[23] Golbeck, J.: Computing and applying trust in web-based social networks, Ph.D. Thesis, University of Maryland (2005).
[24] Shetty, J. and Adibi, J.: Discovering important nodes through graph entropy the case of Enron email database, *Proc. 3rd International Workshop on Link Discovery*, pp.74–81 (2005).
[25] Choudhury, M.D., Mason, W.A., Hofman, J.M. and Watts, D.J.: Inferring relevant social networks from interpersonal communication, *Proc. 19th International Conference on World Wide Web* (*WWW '10*), pp.301–310 (2010).
[26] Rowe, R., Creamer, G., Hershkop, S. and Stolfo, S.J.: Automated social hierarchy detection through email network analysis, *Proc. Joint 9th WebKDD and 1st SNA-KDD Workshp on Web Mining and Social Network Analysis* (*WebKDD/SNA-KDD '07*), pp.109–117 (2007).
[27] Williamson, D.F., Parker, R.A. and Kendrick, J.S.: The box plot: A simple visual method to interpret data, *Annals of Internal Medicine*, Vol.110, No.11, pp.916–921 (1989).
[28] Buckland, M. and Gey, F.: The relationship between recall and precision, *Journal of the American Society for Information Science*, Vol.45, No.1, pp.12–19 (1994).
[29] Dabbish, L.A., Kraut, R.E., Fussell, S. and Kiesler, S.: Understanding email use: Predicting action on a message, *Proc. ACM Conference on Human Factors in Computing Systems* (*CHI '05*), pp.691–700 (2005).

Sho Tsugawa received his M.E. degrees in the Information and Computer Sciences from Osaka University in 2009. He is currently a Ph.D. candidate Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of social network analysis. He is a student member of IEEE, IEICE, and IPSJ.

Hiroyuki Ohsaki received his M.E. degree in the Information and Computer Sciences from Osaka University, Osaka, Japan, in 1995. He also received his Ph.D. degree from Osaka University, Osaka, Japan, in 1997. He is currently an Associate Professor at Department of Information Networking, Graduate School of Information Science and Technology, Osaka University, Japan. His research work is in the area of traffic management in high-speed networks. He is a member of IEEE, IEICE, and IPSJ.

Makoto Imase received his B.E. and M.E. degrees in information engineering from Osaka University in 1975 and 1977, respectively. He received his D.E. degree from Osaka University in 1986. From 1977 to 2001, he was engaged Nippon Telegraph and Telephone Corporation (NTT). He has been a Professor of Graduate School of Information Science and Technology at Osaka University since 2002. His research interests are in the area of information networks, distributed systems and graph theory. He is a member of IPSJ, JSIAM, and IEICE.