Towards Building Effective Email Recipient Recommendation Service

Qi Hu, Shenghua Bao, Jingmin Xu, Wenli Zhou, Min Li, Heyuan Huang
IBM Research - China
Email: {huqihq, baoshhua, xujingm, wlzhou, minliml, huanghey}@cn.ibm.com

***Abstract*—Email is one of the most essential IT services for modern enterprises. However, missing desired recipients for sent messages that frequently happens results in great communication confusion and collaboration inefficiency. In this paper, we propose an effective Email recipient recommendation service implemented by the participant co-occurrence social network based method to help Email users find the missing recipients. Given a specific task of recommending recipient we studied three key factors based on the social network: 1) a general recipient recommendation algorithm based on the social network; 2) the method for updating the edge weight of the social network for the new query (message); and 3) the method to measure the correlated vertices by the social metric of closeness centrality. An extensive evaluation has been conducted on the Enron Email Corpus and Lotus Notes Email Corpus. Experimental results show that the proposed method with a proper estimation of edges and vertices in the social network can achieve significant performance improvement for finding the missing recipients. Five real missing cases of Enron Email Corpus further verify the effectiveness of the proposed method. Besides, we implemented the service in IBM Lotus Notes and invited dozens of colleagues to join in a pilot. The initial feedbacks received from participants are very positive.**

I. Introduction

Email service is one of the most popular collaboration services, in which several people involved in one mail at the same time is very prevalent. The analysis of Gmail shows that over 10% of emails are sent to more than one recipient, and over 4% of emails are sent to 5 or more recipients. The analysis of the email network of Google employees shows that over 40% of emails are sent to more than one recipient, and nearly 10% are sent to 5 or more recipients [1].

Simultaneously, accidently forgotten recipients happen frequently in Email service. Based on the analysis on the Enron Email Corpus, Carvalho *et al.* [2] found that at least 9.27% of the users have forgotten to add a desired email recipient in at least one sent message, while at least 20.52% of the users were not included as recipients (even though they were intended recipients) in at least one received message. So email recipient recommendation service has great value for enterprise communication and collaboration efficiency.

Some preliminary studies have been conducted for personal email recipient recommendation. Pal and McCallum [3] proposed graphical model based on the body words, subject words as well as the current recipients. Carvalho *et al.* [2], [4], [5] investigated how to enhance the prediction for recipient recommendation by several factors such as recency and content. They combined these factors together using technologies of data fusion and machine learning. Roth *et al.* [1] proposed friend suggest algorithm to predict the recipients by implicit social graph, which is a group based solution and without considering the content of the message. These solutions advance the recipient recommendation task to some extent. However, all of them do not consider the indirect/transitive relationship between each pair of the participants in the message which has great impact on finding the missing recipients. In addition, the computation of some key factors, like term based content analysis and exponential delay based time factor, need to be further investigated.

In this paper, an Email recipient recommendation service implemented by the participant co-occurrence social network based method is proposed to help email users to find the missing participants for current email conversation. Given a specific task of recommending recipient we studied three key factors based on the social network: 1) a general recipient recommendation algorithm including offline initializing social network, online social network analysis, and multiple evidence-based rank aggregation; 2) the method for estimating and updating the weights of edges in the social network. The impact of both recency and topical similarity are carefully simulated according to human cognition; and 3) the method to measure the correlated vertices under a set of pre-specified vertices in the social network by the social metric of closeness centrality.

An extensive evaluation on both the Enron Email Corpus and Lotus Notes Email Corpus verifies the effectiveness of the proposed method. On the Enron Email Corpus, the experimental results show that by taking the transitive relationship into account, the proposed method has significant improvement compared with two baselines. This improvement keeps consistent at different size of pre-specified recipients. Five real missing cases in Enron dataset further verify the effectiveness of the proposed method. On the Lotus Notes Email Corpus of our own mailbox, the performance of the proposed method is also outstanding. Useful as it is, the proposed model has been deployed in IBM Lotus Notes and receives very positive feedbacks.

The rest of the paper is organized as follows. Section II introduces the related work. Section III states the method of recommending email recipient within social network. In Section IV, the evaluation on the Enron Email Corpus and Lotus Notes Email Corpus is depicted. Section V shows an application of a plug-in in IBM Lotus Notes to provide Email

978-1-4673-2401-4/12/$31.00 ©2012 IEEE

recipient recommendation service. Finally, some concluding remarks and future work in Section VI.

II. Related Work

Recipient recommendation problem was initially proposed by Pal and McCallum [3] which they called CC prediction. They extended the standard Naive Bayes model [6] to graphical models leveraging the body words, subject words and the current recipients extracted from the message. They utilized both the content and recipients of the message, but did not utilize the time factor of the message which is also a very important feature for messages.

Carvalho *et al.* [2], [4], [5] did several works on the task of recipient recommendation on the Enron Email Corpus. In [4] they formalized the task as a large-scale multi-class classification problem and used the confidence of the classifier to rank the recipients. In [2], they treated the problem as intelligent message addressing for user-ranking, compared several models including adaptation of formal expert search models [7] and classification-based models [8], and used data fusion [9] method to combine the evidence of the models. In [5], they implemented their algorithm of recipients ranking by combining the rank results of recency, frequency and content using data fusion techniques based on the Mean Reciprocal Rank [10], as a plug-in for Thunderbird[1] named *CutOnce*. All these methods are to rank the recipients by combining all kinds of factors (both content and network-based features) using classifier or data fusion technologies. However, they did not consider the relationship between participants (sender and recipients) and analyzed the content of the message based on term analysis, which can be further improved.

Without considering content, Roth *et al.* [1] focused on user interactions and used implicit social graph to address this problem. The implicit social graph was formed by users' interactions with contacts and groups of contacts. They proposed a suggesting friend algorithm and several methods to compute the score for each group. Their method is a group based solution which is not convenient to analyze the relationship between each pair of participants, especially the transitive relationship. Furthermore, this method does not consider the content factor.

III. Recommending Recipients within Social Network

In this section, the method for recommending recipient in participant co-occurrence social network is introduced. A recipient recommendation algorithm is described to illustrate the core routine of the method.

A. Recipient Recommendation Algorithm

The core routine of our method is shown in Algorithm 1. Similar as [1], pre-specified recipients in the new message are defined as *seed* of the recipient recommendation task.

The first step of recommending recipients within social network is to initialize the social network using the messages in the email history. Then for each query (new message), the weight of the network is computed according to the *timestamp* and *content* of the query respectively, and the vertices are measured according to the *seed* recipients pre-specified in the query. Finally, the ranking results of content computation and recency computation are aggregated.

Algorithm 1 Recommend Recipients within Social Network

Offline Initialization of Social Network:
1. Build social network with the email history
  1.1 Extract the vertex set from sent and received messages
  1.2 Extract the edge set for each pair of vertices from each message involving both of them

Online Recommendation with Social Network Analysis:
2. Compute the rank of the candidate vertex, for each email query (*timestamp*, *content*, *seed*)
  2.1 Compute the weight of the edge with *timestamp* and *content* of the email query separately
  2.2 Measure and rank the vertex by the social metric according to *seed* recipients of the email query
3. Aggregate the ranking results of different weighting computation methods

B. Social Network Initialization

Before recommending recipients for each new message, the messages in the email history were used to initialize the social network. The participant co-occurrence social network is defined as an undirected graph $G = (V, E)$ where the edges are weighted.

- The vertex set $V = \{v_1, v_2, \ldots, v_n\}$ is the participant set whose corresponding email addresses appeared in the messages. The i-th participant is denoted by v_i.
- The edge set $E = \{e_1, e_2, \ldots, e_m\}$ records the participant co-occurrence information. An edge $e_i = v_j v_k$ is an element of edge set E if and only if v_j and v_k appeared together in at least one message.

The vertices in set V of the network are extracted from the sent messages and received messages, which are the participants whose email addresses involved in the messages. An edge linking two vertices is added to edge set E if and only if the two corresponding recipients co-occur in the same message. The weight of the edge depicts the relevance of the two vertices under pre-specified vertices, which is computed by weighted summing the number of messages involving both of the participants, and weighting each message by a score, which is illustrated in the next section.

C. Weighting Social Network

The weight of the social network is computed with two parameters acquired from the new message: one is the timestamp of the new message; the other is the content of the new message. Here, the weight of the social network based on the time factor and content factor is computed separately.

Recency Computation When considering the time effect of the message, we analogy it with the memories. Assuming

[1] http://www.mozillamessaging.com/thunderbird/

the message is sent and never read by the user any more, the message is slipping of the memory as the time elapses. Therefore, the time effect of the message depends on the memory for this message.

As a pioneered work on the experimental study of memory, Ebbinghaus curve [11] on behalf the human understanding for the memories. In order to find approximate function expression for memory, we use power function $a \cdot x^b$ and exponential function $a \cdot e^{bx}$ to approach the Ebbinghaus curve. The result is that compared with exponential function, power function is more fitting. Therefore, the power function is selected to measure the recency of message. Assuming the timestamp of the new message m_{new} is $t(m_{new})$, the recency score of each message m_i is calculated by the power function, which is shown in (1) :

$$S_{rcy}(m_i) = (t(m_{new} - t(m_i)))^{-\lambda} \quad (1)$$

where $t(m_i)$ presents the timestamp of the message $m_i \in M_s \cup M_r$, M_s and M_r present the sent messages and received messages in the email history respectively. λ is the smoothing factor for the power function.

Different from [1] and [2], which measuring the recency of the message by exponential decay function (using $(1/2)^x$ in [1] and using $(1/e)^x$ in [2]), we calculate it by power function which is more approach to the memory curve. Comparing with exponential decay function, power function is more flat. Such a property ensures the effect of recency in our method is not so prominent and frequency has more impact on the measurement.

After getting the recency score of each message, the next step is to compute the weight of the social network. For participants v_i and v_j in the social network, $w_r(v_iv_j)$ which is the weight of the edge v_iv_j is calculated as follows:

$$w_r(v_iv_j) = \omega \sum_{m \in M_s(v_iv_j)} S_{rcy}(m) + \sum_{m \in M_r(v_iv_j)} S_{rcy}(m) \quad (2)$$

where $M_s(v_iv_j)$ presents the sent messages set, in which the messages contain both v_i and v_j. Similarly, the $M_r(v_iv_j)$ presents the received messages set, in which the messages contain both v_i and v_j. ω presents the relative importance of sent messages versus received messages. $S_{rcy}(m)$ presents the recency score of message m, which is computed by (1).

Content Computation Content is very important information in email network. Compared with the term analysis, we prefer using topic analysis to analyze the content of the message because the topic model is a better refinement of the content than the term space model by alleviating the term mismatch problem.

Based on current topic models, such as LDA (Latent Dirichlet Allocation) [12], the message is mapped to a mixture of various topics with corresponding probability. In this case, the topic information for each message m_i is represented as a vector, say $\vec{c}(m_i)$. Assuming there are j topics,

$$\vec{c}(m_i) =< p_1^{m_i}, \ldots, p_j^{m_i} >, \sum_{k=1}^{j} p_k^{m_i} = 1 \quad (3)$$

where $p_k^{m_i}$ present the probability of message m_i belonging to the topic k. Similarly, the topical vector of the new message m_{new}, $\vec{c}(m_{new})$ can be represented as follows:

$$\vec{c}(m_{new}) =< p_1^{m_{new}}, \ldots, p_j^{m_{new}} >, \sum_{k=1}^{j} p_k^{m_{new}} = 1 \quad (4)$$

where $p_k^{m_{new}}$ represents the probability of the new message m_{new} belonging to the topic k.

Then, the topical similarity of each message m_i with the new message m_{new} can be defined as the cosine similarity of these two vectors, which is shown in (5).

$$S_{cnt}(m_i) = \frac{\vec{c}(m_i) \bullet \vec{c}(m_{new})}{|\vec{c}(m_i)| \times \vec{c}(m_{new})} \quad (5)$$

Similar as recency computation, the weight computation by content similarity is shown in (6). For recipients v_i and v_j in the social network, $w_c(v_iv_j)$ which is the weight of the edge v_iv_j is calculated as follows:

$$w_c(v_iv_j) = \omega \sum_{m \in M_s(v_iv_j)} S_{cnt}(m) + \sum_{m \in M_r(v_iv_j)} S_{cnt}(m) \quad (6)$$

where $M_s(v_iv_j)$ presents the sent messages set involving both v_i and v_j. Similarly, the $M_r(v_iv_j)$ presents the received messages set. ω presents the relative importance of sent messages versus received messages. $S_{cnt}(m)$ presents the content score of message m, which is computed by (5).

D. Social Metric

In order to rank each vertex in the social network, the next step is to measure the social correlation of the vertex and *seed*. In this paper, we mainly consider the social metric of *Closeness centrality*. *Closeness centrality* considers the transitive relationship between the *seed* and each vertex. The correlation between each pair of vertices is calculated by the shortest distance between two vertices. Here, the candidate vertices are the set of vertices that are connected with *seed* in the network, which brings more possibility to find the missing recipients.

Formally, for vertex v_i, $ClosenessCent(v_i)$ is defined as the inverse of normalized distance sum between *seed* and the vertex v_i.

$$ClosenessCent(v_i) = \frac{|seed|}{\sum_{v_j \in seed} d(v_i, v_j)} \quad (7)$$

where $d(v_i, v_j)$ is the shortest distance of vertex v_i and vertex v_j when the two vertices are connected. If there is no connection between v_i and v_j, then $d(v_i, v_j) = D_{max}$ where D_{max}=$max\{d(v_i, v_j)|v_i$ and v_j are connected in $G\}$. $|seed|$ presents the number of the vertices in *seed*.

Note that the weight of the edge v_iv_j presents the relevancy of vertex v_i and vertex v_j. The distance should be calculated by the inversed weight $w'(v_iv_j)$ of the edge v_iv_j in (8). $w(v_iv_j)$ is the weight of the edge v_iv_j, C_{max} is a constant presenting the maximum weight of the edges in the network.

$$w'(v_iv_j) = C_{max} - w(v_iv_j) \quad (8)$$

E. Ranking Aggregation

Recency and topical similarity of the message are two features independent with each other. However, the distribution of recency score and distribution of content score are quite different that they are not in the same order of magnitude. Therefore, it is hard to fusion the two features in weight computation. Here, the rank result $Rank_{recency}(v_i)$ of recency computation and rank result $Rank_{content}(v_i)$ of content computation are combined based on Mean Reciprocal Rank [9], [10].

For each candidate vertex v_i, the rank of the candidate $Rank(v_i)$ is computed as the weighted reciprocal sum of $Rank_{recency}(v_i)$ and $Rank_{content}(v_i)$, which is shown in (9). α is the parameter used to adjust the relative impact of recency and content.

$$Rank(v_i) = \frac{\alpha}{Rank_{content}(v_i)} + \frac{1-\alpha}{Rank_{recency}(v_i)} \quad (9)$$

IV. Evaluation

In this section, the Enron Dataset as well as the first author's mailbox are selected to evaluate the method. Recall that Section 2 discusses group based solution [1] proposed by *Google* and *CutOnce* [5] proposed by *CMU* (Carnegie Mellon University), which are both classic solutions on the problem of email recipient recommendation. Both of them are selected as the baselines called ***Google-SuggestFriend*** and ***CMU-CutOnce***. And the method proposed in this paper is named as ***PCSN***, which is short for *Participant Co-occurrence Social Network*.

A. Experimental Setup

The method is evaluated on the Enron Email Corpus which is a large collection of real emails mostly sent along the year of 2000-2002. The version of the Enron Email Corpus chosen to use is compiled by Jitesh and Adibi [13]. Before using the dataset to do the experiment, some preprocessing is done including deal with the multiple email address problem, remove the duplicate messages, filter out the noisy content of the messages and select the messages with 2-25 recipients.

For each Enron user, two sets of messages are constructed: sent collection and received collection. Similar as [2], both the sent and received collection are sorted chronologically, and the sent collection is split into the training set and the testing set. Here, 30% of the sent collection is chosen as the testing set.

Also each user's address book is simulated, which containing all email addresses extracted from the TO, CC or BCC field of the sent collection and received collection in the training set. Table I shows the statistic results for candidate recipients, training messages and testing messages of the 10 selected Enron users.

B. Experimental Results

In this part, the experimental results of the method is illustrated by quantitative evaluation and effectiveness analysis. The *seed*-based method is used to evaluate the accuracy of the prediction of the recipient recommendation, the same as [1]. The experiments with *seed* size from 1 to 3 have been done. Due to the space constraint, only detailed analysis on results with the *seed* size 2 is mentioned here. Three measures in evaluation are used here, including *MAP*, *R-precision*, and *Top N precision* (*P@N*). The results of the metrics are implemented by standard TREC evaluation tool[2].

TABLE I
STATISTIC RESULTS FOR 10 SELECTED ENRON USERS

	Candidate Recipients	Training Messages	Testing Messages
Mean	429.20	417.90	38.50
StDev	179.48	124.22	16.32
Max	919	744	63
Min	308	303	18

Quantitative Evaluation Table II shows the experimental results of the three methods. It is easy to find that *PCSN* achieves the best performance, which significantly outperforms *CMU-cutonce* and *Google-SuggestFriend* by 67.6% and 41.2% on *MAP*, respectively.

TABLE II
EXPERIMENTAL RESULTS COMPARED WITH BASELINES

Methd	MAP	R-Prec	P@5	P@10
CMU-CutOnce	0.272	0.193	0.178	0.140
Google-SuggestFriend	0.323	0.231	0.191	0.147
PCSN	**0.456**	**0.378**	**0.286**	**0.194**

In addition, recall/precision curve of the three methods at *seed* size 2 is shown in Fig. 1. As we can see, the proposed method *PCSN* consistently achieves the best performance.

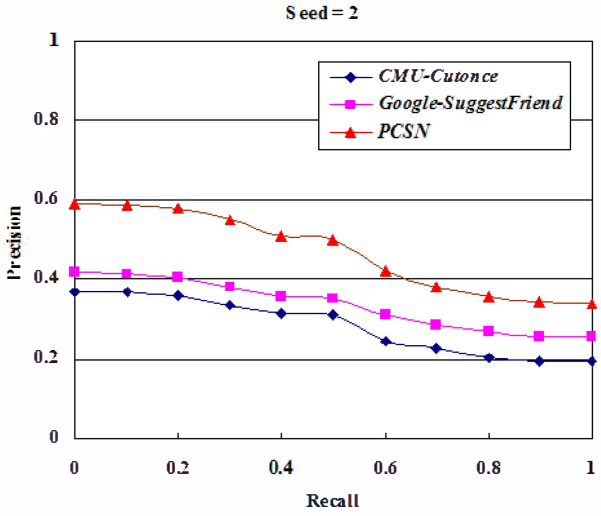

Fig. 1. Recall/Precision curve at seed size 2

In order to illustrate the performance of the method comprehensively, the performance at different *seed* sizes is shown. Table III shows the *MAP* results of the three methods with *seed* size ranging between 1 and 3. As we can see the proposed method of *PCSN* keeps high accuracy of predicting remaining

[2]http://trec.nist.gov/trec_eval/

recipients of the message, given different size of recipients pre-specified by the sender. Again, the proposed method constantly achieves the best performance among the three methods at different *seed* sizes.

TABLE III
RECOMMENDATION RESULTS WITH DIFFERENT SEED SIZES

Methd	Seed=1	Seed=2	Seed=3
CMU-CutOnce	0.259	0.272	0.248
Google-SuggestFriend	0.333	0.323	0.299
PCSN	**0.399**	**0.456**	**0.426**

Effectiveness Analysis In this part, the effectiveness of the method is further analyzed in two aspects: effectiveness of aggregation method compared with only content method and only recency method, and the parameter selection for the method.

In order to illustrate the effectiveness of aggregation method, experiments for each selected users with three methods of *PCSN*, *PCSN (recency)* and *PCSN (topic)* are done here. *PCSN (recency)* presents the method of only considering time factor while *PCSN (topic)* presents the method of only considering content factor. The result is that rank aggregation is much more stable than that based on only content or recency, which is shown in Fig. 2.

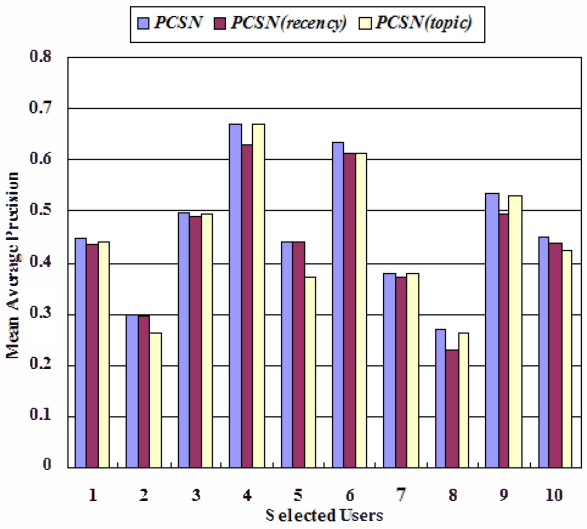

Fig. 2. MAP result for 10 selected users

Fig. 2 depicts that the performance of content computation method and that of topic computation method are different for different users: for some users, the performance of content computation method is better than that of recency computation, while for some users, the result is reversed. However, for all the selected users, the method of *PCSN* outperforms other two methods, which presents that the method of considering both content and recency factors has stable and better performance than that considering only one of the two factors.

For the method of *PCSN*, there are three parameters: α, λ and ω. Experiments are done for each parameter with other two parameters fixed. The best performance shows when $\alpha = 0.6$, $\lambda = 1.5$, $\omega = 6$. And the experimental result shows that the proposed method is not sensitive to any of these three parameters. Therefore, the proposed method has good practicality for different users.

C. Case Study: Finding Real Missing Recipients

In previous section the proposed model is evaluated by selecting one or more recipients as the *seed* to find other recipients in the recipient list based on Enron Email Corpus. In this section the model is evaluated with real missing cases, in which the *seed* containing all recipients in the original message and the missing recipient has been identified based on some analysis.

According to the description of Carvalho *et al.* [2], some candidate missing recipients cases are found in the Enron Email Corpus by searching the messages containing the key word like sorry, forgot and accident. For these candidate cases, we check them one by one manually and select 5 real missing cases. The real missing cases are judged by containing some sentences such as "I forgot to include you on the cc" and "forgot to send it to you as well".

The (*content*, *timestamp*, *recipients*) of the message which the real missing case happened is extracted. The messages before the *timestamp* in the sender's message box are chosen as the training data. Taking *recipients* as *seed*, we build the model of *PCSN* for the missing case. The results of found real missing recipients are shown in Table IV. It is easy to find that the performance of the model is promising in finding real missing recipients. For these five cases, the average precision at 1 is 0.6 and the *MAP* is 0.628. Therefore, there is high possibility for the proposed model to recommend the missing recipients at the top list.

TABLE IV
REAL MISSING RECIPIENTS CASES IN ENRON EMAIL CORPUS

Missing Case (MID)	Missing Recipients	MAP	P@1
237491	sh*****.daniel@enron.com	1	1
239524	ge*****.nemec@enron.com	1	1
242117	de****.lagesse@enron.com	0.024	0
260010	rh****.denton@enron.com	1	1
9930	an**.koehler@enron.com	0.11	0

D. Experiment on Lotus Notes Email Corpus

As an unique real large email dataset, the Enron Email Corpus is commonly used in email research. In this section, Lotus Notes email corpus is used as another real dataset to evaluate the model.

The first author's mailbox is selected as the experiment dataset. There are 1870 messages in the mailbox, including 573 sent messages and 1297 received messages. After filtering the messages, there are 628 training messages and 49 testing messages. The method of *PCSN* is implemented with parameter $\alpha = 0.6$, $\lambda = 1.5$, $\omega = 6$ which is the same parameter as implemented in the Enron dataset. For $seed = 2$, the result of *MAP* is 0.620 and the result of *P@5* is 0.478. The result

depicts that the proposed method has good performance on the real mailbox dataset.

The results on both Enron Email Corpus and our own local mails depict that the proposed method can achieve similar performance when domain changes. In other words, the proposed method is domain independently and has wide applicability.

V. Application

The model of *PCSN* has been implemented as a plug-in to deploy in IBM Lotus Notes to provide Email recipient recommendation service. As shown in Fig. 3, this plug-in has three features:

- Recommend missing recipients. According to the pre-specified recipients and content of the composed mail, the potential missing recipients will be recommended to the Email user. In the dialogue, the user can select one or more email addresses in the left table and click "To/CC/Bcc" button to add the email address to the recipient list in the right tree.
- Dynamically updating. Each time the recipient list is changed by the user's edition, the recommendation list will be updated. The response time for this updating is around 100 milliseconds.
- Cancel send. This feature is not complex but provides good user experience for users if he/she wants to re-edit the message before sending out.

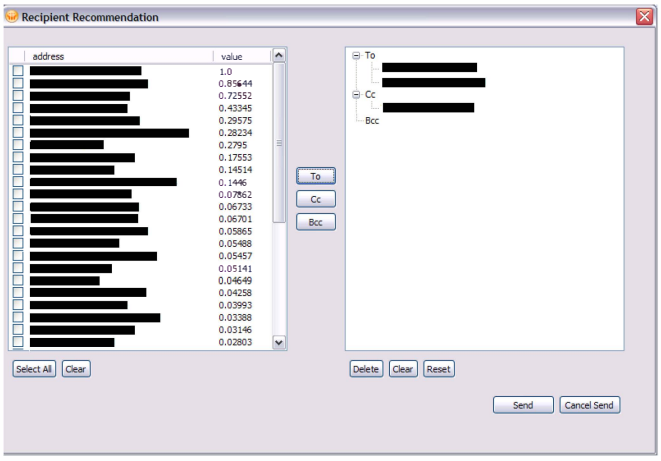

Fig. 3. Recipient recommendation plugin in IBM Lotus Notes

Now dozens of colleagues have been invited to join in a pilot, who have using this application for about several weeks. The users' behaviors (e.g. how many users in the list are selected and what is the rank of the user, etc) are observed and recorded. Currently, we get their initial feedbacks by questionnaire, which is shown in Table V.

TABLE V
User feedbacks for IBM Lotus Notes plugin

Like
1. The quality of the suggested recipient is good
2. Quick response
3. Provide the second chance to confirm the recipient list
To be improved
1. Introduce a recommendation confidence threshold and show up the recommendation window only if necessary, e.g., the recipient recommendation confidence is higher than the threshold.

VI. Conclusion and Future Work

Missing important participants in email conversions happens frequently, which results in inefficient collaboration. Email recipient recommendation service is a valuable addition for the email application. In this paper, we focus on dealing with the problem of email recipient recommendation using social network analysis. A method is proposed to do recipient recommendation based on the participant co-occurrence social network. We design a general recipient recommendation algorithm within social network. Extensive evaluation on Enron email corpus and IBM Lotus Notes corpus verifies the effectiveness of the proposed method.

As for future work, we plan to further enhance the performance of the recommendation by trying new social metrics for measuring the vertex in the network. Moreover, we will try to use the similar technology to prevent email information leaks, i.e., when a message is accidentally addressed to non-desired recipients. In this case, only the already-specified recipients of the message need to be ranked (with the least likely recipient on the top).

References

[1] M. Roth, A. B. David, D. Deutscher, G. Flysher, I. Horn, A. Leichtbery, N. Leiser, R. Merom, and Y. Mattias, "Suggesting friends using the implicit social graph," In Proceedings of the 16th ACM SIGKDD, 2010, pp.233-242.

[2] V. R. Carvalho, and W. W. Cohen, "Ranking users for intelligent message addressing," In Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval, 2008, pp.321-333.

[3] C. Pal, and A. McCallum, "CC prediction with graphical models," In the third Conference on Email and Anti-Spam (CEAS), 2006.

[4] V. R. Carvalho, and W. W. Cohen, "Predicting Recipients In the Enron Email Corpus," Technical Report CMU-LTI-07-005, 2007.

[5] R. Balasubramanyan, V. R. Carvalho, and W. W. Cohen, "Cutonce - recipient recommendation and leak detection in action," In Proceedings of EMAIL-08: the AAAI Workshop on Enhanced Messaging, 2008.

[6] A. McCallum, and K.Nigam, "A comparison of event models for naive bayes text classification," In Proceedings of AAAI-98 Workshop on Learning for Text Categorization, 1998, pp.41-48.

[7] K. Balog, L. Azzopardi, and M. D. Rijke, "Formal models for expert finding in enterprise corpora," In Proceeding of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2006, pp.43-50.

[8] Y. Yang, and X. Liu, "A re-examination of text categorization methods," In Proceeding of 22nd Annual International SIGIR, 1999, pp.42-49.

[9] J. A. Aslam, and M. Montague, "Models for metasearch," In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2001, pp.276-284.

[10] C. MacDonald, and I. Ounis, "Voting for candidates: adapting data fusion techniques for an expert search task," In Proceedings of the 15th ACM International Conference on Information and Knowledge Management, 2006, pp.387-396.

[11] H. Ebbinghaus, "Memory: A Contribution to Experimental Psychology," Teachers College, Columbia University, 1913.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," Journal of Machine Learning Research, 2003, Vol.3, pp.993-1022.

[13] J. Shetty, and J.Adibi, "Enron Email Dataset," Technical report, USC Information Sciences Institute, http://www.isi.edu/ adibi/Enron/Enron.htm, 2004.