Email Recipient Recommendation

A Survey of Current Work and My Approach

Yifu Huang
Software Engineering Institution
East China Normal University
Shanghai, China
10092510437@ecnu.cn

***Abstract*— Email recipient recommendation suggests who recipients of an email might be, while the email is being composed, given its current contents and given its previously-specified recipients. It can be a valuable addition to both personal and corporation's email clients. Email recipient recommendation can be used to identify people in an organization that are working in a similar topic or project, or to find people with appropriate expertise or skills. It can also prevent a user from forgetting to add an important collaborator or manager as recipient, preventing costly misunderstandings and communication delays. In this paper, I examine the current work in graphic models and multi-class classification, and propose my approach to it with machine-learned ranking. Finally, I point out the limitation of current work and the future direction with my consideration.**

Keywords-email, recipient recommendation, ranking.

I. INTRODUCTION

Email recipient recommendation suggests who recipients of an email might be, while the email is being composed, given its current contents and given its previously-specified recipients. It is considered as an automated technique, which is designed to avoid a specific type of high-cost email error: errors that result when a message is not sent to all intended recipients. An example of such an error would be forgetting to CC an important collaborator, or manager, on a message to a working group: such an omission could cause costly misunderstandings and communication delays. With the help of email recipient recommendation, user can avoid above-mentioned errors and improve work efficiencies.

The current work of email recipient recommendation is mainly based on classification. Only two teams have discussed this problem as far as I know, and they propose their methods respectively. First, Chris Pal and Andrew McCallum [1] first address this problem using graphical models. Second, Vitor R. Carvalho and William W. Cohen [2] formalize this task as a large-scale multi-class classification problem.

In this paper, I examine the current work of email recipient recommendation in detail, and discuss the advantages and limitations respectively. Then I propose my approach to it with machine-learned ranking. Finally I point out the future direction with my consideration.

II. GRAPHIC MODELS

Chris Pal and Andrew McCallum first address this problem using graphical models for words in the body and subject line of the email as well as the recipients given so far on the email. They consider that recipient recommendation is closely related to the problem of expert finding in an organization. And they present results using naively structured models and introduce a powerful new modeling tool: plated factor graphs.

A) Data Set

The email data set that they use is one of the author's. It contains 825 unique users and 9244 sent mailbox emails during the nine-month period in 2004.

B) Methods

In their work there they begin with a simple multinomial Naïve Bayes model for words in the body of the message under composition. In this model they focus on sent mail box only, and consider each recipient as a target label y and replicate emails where necessary. Due to different numbers of words n for any possible email, they simply instantiate n observed words for an email under composition, use the model to compute the distribution over labels y, and present the user with a list sorted by its probability.

Their other work focuses on extending the standard Naïve Bayes. In their approach there, they partition the emails into three different sections, the body, the subject and the recipients. They then use three different discrete conditional distributions for variables observed within these different sections. They process email addresses into a bag of words breaking at periods, spaces and @. This gives them some tolerance for minor perturbations of email addresses when identity resolution is inexact. They propose illustrating these types of models using plated factor graphs that allow mixtures of undirected and directed graphical models to be compactly illustrated.

Their models are in Figure 1.

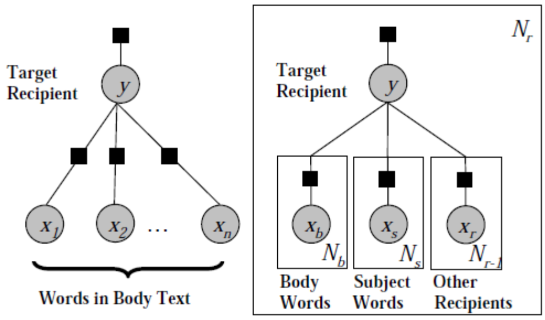


Figure 1. (Left) A factor graph for a Naive Bayes model for email recipient recommendation. (Right) A plated factor graph for a naive model employing different alphabets for words in the body, words in the subject line and for recipients.

C) Evaluation

They begin by estimating the parameters of their models on the first week of email and then re-estimate the models each day at 4:00am. They score email recipient recommendation as correct if the held out recipient is contained within a list of the top five recipient predicted by sorting the probabilities obtained from the model. Their results are provided in Table I.

Model	First Month	Last Month	Avg. Daily
Naive Bayes	.301	.326	.364
Factor Graph	.364	.395	**.448**
Thread Info	.357	.403	.448

TABLE I. A COMPARISION EMAIL RECIPIENT RECOMMENDATION ACCURACY FOR NAIVE BAYES MODELS AND PLATED FACTOR GRAPH MODELS.

D) Advantages

a) Using factor graphs with locally normalized factors for their experiment here so that parameter estimation amounts to computing sufficient statistics which can be quickly computed.

b) This also leads to fast incremental estimation which is an important design criterion that enables a system to rapidly adapt as new email is generated.

c) Another advantage of using the plated factor graph notation is that they can describe models that are not locally normalized as well as models that are obtained via discriminative optimization as is done in the Conditional Random Field (CRF) framework.

d) Their bag of words representation for addresses therefore has the potential to capture some of these address variations.

E) Disadvantage

a) The email data set is personal, so it cannot give general performance evaluation.

b) The evaluation criterion used is too simple to reveal more details of performance.

c) Some important feature such as frequency, recency and co-occurrence of email addresses in the training set are ignored to take into consideration.

d) Standard gradient based optimization methods for the analogous multinomial logistic regression models defined by these graphical structures were unacceptably slow.

F) My Opinion

When I picked up the idea that applying machine learning to email recipient recommendation, I found that Chris Pal and Andrew McCallum had already discussed about it in 2006. It is laudable. They propose bag of words representation for addresses, and it can help weight same domain and similar address. It is in accordance with truth. But their data set is not convictive, and their model ignored some important features.

III. Multi-class Classification

Vitor R. Carvalho and William W. Cohen present the first study of recipient recommendation in a real large-scale corporate email collection, the Enron Email corpus. They begin by defining the problem as a large multi-class multi-label classification task, where each email can be addressed to multiple recipients in the user’s address book. They propose various baselines to the problem, along with a classification-based re-ranking scheme to combine two types of features: textual contents and network information from the email headers.

A) Data Set

The email data set that they use is the Enron dataset version compiled by Jitesh and Adibi [3], in which a large number of repeated messages were removed. This version contains 252,759 messages from 151 employees distributed in approximately 3000 folders. For each Enron user, they considered two distinct sets of messages: both sent collection and received collection. Train and test data are organized as Figure 2.

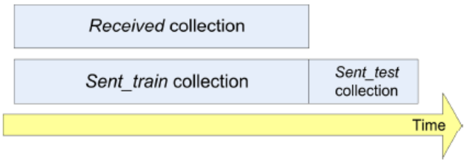


Figure 2. Time frames of different email collections.

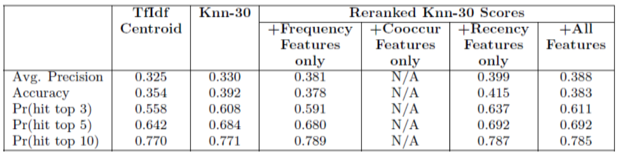
B) Methods

They develop baseline methods taking advantage of the textual information inside email messages. They start by proposing a technique based on cosine similarity between two TF-IDF vector-based representations of email messages. The second method was based on the K-Nearest Neighbors algorithm described by Yang & Liu [4].

In addition to the textual scores, they used three different sets of network features. The first set is based on the relative frequency of a recipient’s email address in the training set. The second type of co-occurrence-based feature is called Relative Joint Recipient Frequency. The last set of network-based features uses the information in the latest messages sent by the user.

C) Evaluation

To evaluate performance, they use well known metrics such as average precision, accuracy and average recall versus rank curves. They test their models on 36 users' emails and their results are in Table II.



	TfIdf Centroid	Knn-30	Reranked Knn-30 Scores			
			+Frequency Features only	+Cooccur Features only	+Recency Features only	+All Features
Avg. Precision	0.325	0.330	0.381	N/A	0.399	0.388
Accuracy	0.354	0.392	0.378	N/A	0.415	0.383
Pr(hit top 3)	0.558	0.608	0.591	N/A	0.637	0.611
Pr(hit top 5)	0.642	0.684	0.680	N/A	0.692	0.692
Pr(hit top 10)	0.770	0.771	0.789	N/A	0.787	0.785

TABLE II. ACCURACIES FOR TFIDF MODELS, KNN-30 MODELS, AND RERANKED KNN-30 MODELS.

D) Advantages

a) A large collection of real email is used to test, since email is a noisy and the email usage varies considerably among users.

b) Those evaluation methods in combination should reveal more details of the task.

c) The re-ranking scheme was shown to be significantly more effective than any of the textual baseline.

d) It can be easily implemented in any email client, not requiring changes in the email server side.

e) Indeed, both the baselines, as well as in the Voted Perceptron-based re-ranking method, are very efficient to train and easy to be implemented in a large-scale systems, especially over an email client that already includes traditional IR search over messages.

E) Disadvantages

a) They have not taken the features of address into consideration, which is proved useful by Chris Pal and Andrew McCallum.

b) Textual features are also mainly disscussed. But it is not in accordance with the truth since people always write recipients before content.

c) Group recommendation is more important than single recommendation, so it should be considered in future.

d) Machine-learned ranking is more appropriate than classification to email recipient recommendation, so it should ben considered in future.

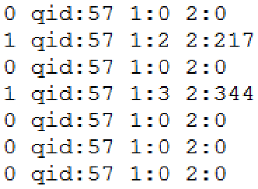
F) My Opinion

Vitor R. Carvalho and William W. Cohen enhanced email recipient recommendation by a large data set, more features, and detail evaluations comparing with Chris Pal and Andrew McCallum's. And they even have put them into practice [5]. It is laudable. But I think machine-learned ranking is more appropriate to email recipient recommendation, and group recommendation is more considerable.

IV. My Approach

In my opinion, I prefer to formalize email recipient recommendation to machine-learned ranking. Machine-learned ranking [6] is a type of supervised or semi-supervised machine learning problem in which the goal is to automatically construct a ranking model from training data. Training data consists of lists of items with some partial order specified between items in each list. This order is typically induced by giving a numerical or ordinal score or a binary judgment for each item, such as "relevant" or "not relevant". Ranking model's purpose is to rank, such as produce a permutation of items in new, unseen lists in a way, which is "similar" to rankings in the training data in some sense.

I get an inspiration from Information Retrieval that how search engine ranks the returned results. In email recipient recommendation, I get train data from sent box and inbox. For each sent message, I consider it as a query. The input of query is the first address of TO and the output of query is all addresses of One's address book ranked. I label latter appeared addresses of TO or CC with "1" for "relevant", and others with "0" for "not relevant". Then I select some feature to every address from every message, such as co-currency that the count of appearing together with the first address, recency that the count of message appearing together with the first address between the most recent one and this message. Then, I use some certain machine-learned ranking tool such as SVM-rank [7] to train ranking model to email recipient recommendation (to get the weight of every feature). Finally, when a message is coming, I will extract feature value of every address and predict priority of all addresses with former model.



```
0 qid:57 1:0 2:0
1 qid:57 1:2 2:217
0 qid:57 1:0 2:0
1 qid:57 1:3 2:344
0 qid:57 1:0 2:0
0 qid:57 1:0 2:0
0 qid:57 1:0 2:0
```

Figure 3. Data format of my approach

During experiment I find that feature selection is very important to my ranking model, and I will add some textual features in the future work, and email group recipients recommendation also merits consideration.

V. Conclusion

In this paper, I examine the current work of email recipient recommendation in detail, and propose my approach to this problem. Chris Pal and Andrew McCallum introduce plated factor graphs to model email recipient recommendation. Vitor R. Carvalho and William W. Cohen formalize this task as a large-scale multi-class classification problem. And my approach is to apply machine-learned ranking model to this problem.

In the future, there should be more models to apply in this problem. To my approach, I will select more valuable features, such as some textual features to enhance the performance of ranking model. And email group recipients recommendation also should be taken into consideration.

References

[1] C. Pal and A. McCallum. Cc prediction with graphical models. *In CEAS*, 2006.

[2] Vitor R. Carvalho and William Cohen. Recommending recipients in the Enron email corpus. *Technical Report CMU-LTI-07-005*, Carnegie Mellon University, 2007.

[3] J. Shetty and J. Adibi. Enron email dataset. Technical report, *USC Information Sciences Institute*, 2004. Available from http://www.isi.edu/~adibi/Enron/Enron.htm.

[4] Y. Yang and X. Liu. A re-examination of text categorization methods. *In 22nd Annual International SIGIR*, pages 42–49, August 1999.

[5] R. Balasubramanyan, V. Carvalho and W. Cohen, CutOnce - Recipient Recommendation and Leak Detection in Action. In AAAI-2008, *Workshop on Enhanced Messaging.*

[6] Machine-learned Ranking wiki page. *http://en.wikipedia.org/wiki/Learning_to_rank*

[7] T. Joachims, Training Linear SVMs in Linear Time, *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.