Email Recipient Recommendation

ACHARI BERRADA Youssef
Master MVA
youssef.achari-berrada@polytechnique.edu

LE SCAON Rém
Master MVA
remi.le-scaon@polytechnique.edu

MOHAJERI Sarah
Master MVA
sarah.mohajeri@polytechnique.edu

SONG Baoyang
Master MVA
baoyang.song@polytechnique.edu

Abstract

This project was done in the framework of the course "Learning for Text and Graph Data" taught by M. Vazirigiannis, during the spring 2017 semester of the MVA Master. The name of our team is **LA_LA Team**.

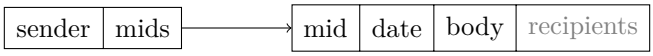
1 Introduction

The goal of this data challenge was to build an email recipient recommendation system based on previous email exchanges within a database. In order to provide recipient recommendations for a given email, we can use not only the recipients, but also the date and contents of the e-mails exchanged between the individuals in the database.
The recipients of the different emails can be used to build the *communication graph*, which models the connections between the users based on their e-mail exchanges. As for the content, it can be processed through text-mining techniques and therefore represent another feature in our recommendations.
In the following, we will describe the different components of our recommendation system and how we used the available data to build them. We will then explain how we tuned the different parameters of our model, and compare the results obtained with our different approaches.

2 Feature Engineering

2.1 Data description

The challenge provides us with a data set containing a main table listing the senders with a list of mail ids "mids" they have sent. For each mid, another info table contains the date, body and recipients of the email. In the training set, the recipients are known, and in the test set, up to ten recipients have to be predicted. See table bellow for an illustration of the dataset.



main table info table

The distribution of the dataset is as follows:

Senders	Mids	
125	Training	Test
	43613	2362

In order to test a given method used for the prediction task, we have created a method called `train_test_split` in order to split the training set into a train and test sets and try the method we want to test.

2.2 Data cleaning

Before feature extraction, we start by cleaning the data provided to us. First, we correct some dates whose year is 000[1-9] instead of 200[1-9]. Then, we replace missing body by empty string ' ' from the info table.
Then we apply to the body of each email the following steps:

- **Lower:** Transform all characters to their lowercases.
- **Tokenization:** Extract words from text without keeping white-spaces and punctuations (we keep the '-' punctuation to preserve intra-word dashes).
- **Pos_tagging:** Keep only words that are nouns or adjectives.
- **StopWords:** Remove English stopwords that have no semantic meaning.
- **Stemming:** Apply Porter's stemming algorithm to reduce tokens to their "root" form.

2.3 Feature extraction

Now our data is ready for feature extraction. In our project, we used two main models and each one is based on different features extracted from the data. The first model is based on TF-IDF representation of the body of the email. And the second model is based on the word count vector representation of the email's body and frequency of the emails exchanged between senders and recipients represented by a graph. We will describe both models and their results in the next section in details.

3 Model Tuning and Comparison

3.1 TF-IDF Centroid

In this approach, we adapt the method developed in [1]. For a given sender s, note $\mathcal{M}_s = \{m_1, \ldots, m_{n_s}\}$ all the messages sent by s in the *training set*. Denote $\vec{\mathcal{M}}_s$ the TF-IDF vectors associated with $\mathcal{M}_s$, then for each user r appears at least once as recipient in $\mathcal{M}_s$, the *TF-IDF centroid* is defined as the sum of all TF-IDF vectors associated emails between s and r, *i.e.*,

$$C_{s,r} = \sum_{m \in \mathcal{M}_s | r \in \text{recipients}(m)} \vec{m}. \tag{1}$$

Notice that, different from [1], each TF-IDF vector $\vec{m}$ in (1) is *not* l^1-normalized.

For testing, an email is first vectorized by the same TF-IDF vectorizer as in training stage and a cosine simularity score is calculated against $\vec{\mathcal{M}}_s$. [1] reranks recipients by this score and reports an average precision **32.5%**. In our experiments, however, only **23.7%** is achieved: we attribute the difference to the different precision measure employed in [1]. Indeed, it is possible that order is not taken into account in their evaluation. Instead, for each sender s, a multi-label SVM is trained. Given a test email from s, a score is produced for each potential recipients and is averaged with the score of baseline method. The recipients are then reranked by the averaged score and we achieved **36.882%** accuracy, which is also our best result in the data challenge.

3.2 Communication Graph and Email Content

In this approach, presented by Graus et al. in [2], the emails sent by all users in the database are used in order to build a *communication graph*, a directed graph where the nodes are the email addresses (senders and receivers), and the arcs represent the email exchanges between two adresses, weighted by the number of emails sent from one user to the other.

This communication graph is then used in order to rank the recipients for an email given its content and the sender, through a generative model.
Using the Bayes' theorem, we can compute the probability of observing a recipient R for an email E sent by S as follows:

$$\mathbb{P}(R|S,E) = \frac{\mathbb{P}(R) \cdot \mathbb{P}(S|R) \cdot \mathbb{P}(E|R,S)}{\mathbb{P}(S) \cdot \mathbb{P}(E|S)} \tag{2}$$

Since we only care about ranking the recipients, we can ignore the normalization factor $\mathbb{P}(S) \cdot \mathbb{P}(E|S)$, and therefore determine the relevance of a recipient for a given email and sender based on the prior probability of the recipient, $\mathbb{P}(R)$, the probability of having this sender given the recipient, $\mathbb{P}(S|R)$, and the probability of the observing this email in the communications between the sender and receiver, $\mathbb{P}(E|R,S)$.

To estimate these three probabilities, we follow the approach of [2]:

1. The prior probability of a recipient, $\mathbb{P}(R)$, is independent from the sender and the email itself. It can be estimated in two different ways, either by considering the number of emails received by R, normalized by the total number of emails sent at that point in time; or by computing R's *PageRank* score in the communication graph as a measure of its relevance as a recipient. This last option was not recommended by the authors, we did not implement it.

2. $\mathbb{P}(S|R)$ is estimated in two different ways.
   - Based on the frequency of the emails S sent to R at that point in time:

     $$\mathbb{P}_{\text{freq}}(S|R) = \frac{n(e, S \to R)}{\sum_{S' \in \mathcal{S}} n(e, S' \to R)} \ , \tag{3}$$

     where $n(e, S \to R)$ is the number of emails sent from S to R.

   - By considering the number of times S and R co-occur as addresses in an email:

     $$\mathbb{P}_{\text{co}}(S|R) = \frac{n(e, \to R, S)}{n(e, \to R) + n(e, \to S)} \ , \tag{4}$$

     where $n(e, \to X)$ corresponds to the number of emails sent to X.

3. Finally, $\mathbb{P}(E|R,S)$ is estimated using the words in the email:

   $$\mathbb{P}(E|R,S) = \prod_{w \in E} \lambda \mathbb{P}(w|R,S) + \gamma \mathbb{P}(w|R) + \beta \mathbb{P}(w), \tag{5}$$

   where $\mathbb{P}(w|\cdot) = \frac{n(w,\cdot)}{|\cdot|}$ is the frequency of the word w in the set of documents. λ, γ and β are parameters with $\lambda + \gamma + \beta = 1$.

We used cross-validation in order to tune the parameters λ, γ and β of our models, and to decide which of the models to use for the estimation of $\mathbb{P}(S|R)$.

Implementation. In order to implement this method, we used the library `igraph`. We created a class `CommunicationGraph` to represent the training set. Each node is affected to a user, and each oriented edge stores a summary of the emails sent from one user to the other, in particular the number of emails, and the sum of word counts in the all these emails. To make the predictions, we computed the scores of each recipient given an email and the sender according to equation (2).

Parameters tuning. To estimate $\mathbb{P}(S|R)$, we compared the performances of $\mathbb{P}_{\text{freq}}$ and $\mathbb{P}_{\text{co}}$, the ferquency approach was selected. The values of the parameters kept for estimating $\mathbb{P}(E|R,S)$ were $\lambda = 0.6$, $\gamma = 0.2$ and $\beta = 0.2$. Moreover, we tried to balance the contributions of the content-related probability with respect to the network-related probabilities by introducing a parameter μ such that $\mathbb{P}(R|S,E) \propto (\mathbb{P}(R) \cdot \mathbb{P}(S|R))^{(1-\mu)} \cdot (\mathbb{P}(E|R,S))^{\mu}$. The results of this cross-validation are presented below, using MAP@10 scores with both true recipients and the ones predicted by the baseline frequency approach. Low μ yields similar predictions to the frequency approach. The best prediction is obtained for $\mu = 0.3$.

μ	MAP (w. true)	MAP (w. freq)
0	0.186	0.922
0.1	0.364	0.552
0.2	0.375	0.434
0.3	**0.388**	0.338
0.4	0.352	0.298
0.7	0.324	0.244
1	0.296	0.188

Result. The score obtained on the leaderboard with this approach did not beat the TF-IDF centroid approach, **0.3606** on the public leaderboard, which is lower than the results obtained by the authors. One point that can explain this difference is the fact that emails that had to be predicted in the challenge were from a different time period as the ones used for training, which was not the case in the paper. Due to the long computation time with `igraph` we did not have time to push further this approach.

4 Going Further and Conclusion

Similar method to the TF-IDF centroid model, was the k-Nearest Neighbors algorithm in which we add weights to recipients based on the similarity between the mail sent and the top k most similar emails selected from the training set, and we compute the weights of recipient r based on the emails (among the top k) in which r is among the recipients like described in [1], section 3.1.2. Our approach focuses on the body of the emails and the relations sender/recipient. We can also exploit the date and more precisely the temporal proximity to compute recency features that improve the kNN method significantly according to [1].

Note: The `readme.txt` provides instructions on how to use the code.

References

[1] Vitor R. Carvalho and William W. Cohen. Recommending recipients in the enron email corpus. Technical report, 2007.

[2] David Graus, David van Dijk, Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. Recipient recommendation in enterprises using communication graphs and email content. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '14, pages 1079–1082, New York, NY, USA, 2014. ACM.