Learning User Embeddings from Emails

Yan Song
Microsoft
One Microsoft Way
Redmond, WA, USA, 98052
yansong@microsoft.com

Chia-Jung Lee*
Microsoft
One Microsoft Way
Redmond, WA, USA, 98052
cjlee@microsoft.com

Abstract

Many important email-related tasks, such as email classification or search, highly rely on building quality document representations (e.g., bag-of-words or key phrases) to assist matching and understanding. Despite prior success on representing textual messages, creating quality user representations from emails was overlooked. In this paper, we propose to represent users using embeddings that are trained to reflect the email communication network. Our experiments on Enron dataset suggest that the resulting embeddings capture the semantic distance between users. To assess the quality of embeddings in a real-world application, we carry out auto-foldering task where the lexical representation of an email is enriched with user embedding features. Our results show that folder prediction accuracy is improved when embedding features are present across multiple settings.

1 Introduction

Email has been an important asynchronous communication channel that people use on a daily basis. A large body of research has laid focus on creating intelligent systems by analyzing the content of email messages, with a purpose to assist users in automating their tasks (Lewis and Knowles, 1997; Drucker et al., 1999; Kushmerick and Lau, 2005; Tam et al., 2012). Email classification, as an example, relies on machine learned models to categorize messages into folders by using text features such as bag-of-words or keywords (Bekkerman et al., 2004; Dredze et al., 2008). Similarly, tasks such as email search (Minkov et al., 2008), email summarization (Carenini et al., 2008), and spam filtering (Gee, 2003) all depend on properly representing the content of the message body, which then can be consumed in the target tasks. While many of these studies have brought success in representing textual messages, creating quality representations of users was not fully investigated.

Considering users as nodes in a graph spanned by email correspondences, a good representation of users can be helpful for many tasks since information is communicated from/to these vertices. In the email domain, the mainstream approaches to representing users are based on bag-of-words or keywords features (Bekkerman et al., 2004; Dredze et al., 2008). Many previous efforts model users and their interactions in social networks or recommendation systems (Grover and Leskovec, 2016; Liang et al., 2016; Zhao et al., 2010). Emails, although can be viewed as a special kind of social platform, tend to generate interactions within a smaller group of participants, requiring a dense representation to help bridge the gap between even the farthest users. In this paper, we propose to learn user embeddings to form such representations, with an aim that these embeddings can bring benefits to email-related tasks.

To learn user embeddings, we consider a graph structure formed by vertices of senders and recipients, which are connected by edges of the messages they exchange. Based on this graph, our approach learns user embeddings jointly with word embeddings in a concatenated space, which treats users as features affecting the semantics of the email content. The resulting user embeddings are expected to correspond to users' sending and receiving activities.

We conduct embedding learning using a publicly available email corpus – the Enron dataset. Our analytical results suggest that the more often users communicate, the more similar their em-

*Both authors contributed equally to this work.

Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, pages 733–738,
Valencia, Spain, April 3-7, 2017.

beddings are. To study the effectiveness of user embeddings in a real application, we apply user embeddings to a surrogate task – email auto-foldering, where lexical and embeddings features are employed for folder prediction. We follow a conventional setting (Bekkerman et al., 2004; Dredze et al., 2008; Tam et al., 2012) where a selected set of users are tested. Our baseline approaches take into account two most effective setups from prior work where the combination of email content and metadata is featurized. Our experimental results show that incorporating user embeddings consistently improves prediction accuracy compared to those with only lexical features.

2 Approach

Our approach to learning user embeddings is based on the continuous bag-of-words (CBOW) structure, similar to the method proposed by Le and Mikolov (2014), which treats paragraph as an external feature that affects, and being trained in, the process of word embedding learning. We take the essence of the aforementioned work, and on the top of that add user embeddings from both sender and recipients to learn word embeddings. Following this design, the semantics captured by word embedding learning are expected to be affected by users who are involved in the email communication.

Figure 1 shows the framework of our approach. The projection layer is a concatenation of user and word embeddings following the order of sender, words and recipients. Since most email scenarios usually involve more than one recipient, our framework averages the embeddings from all recipients in the projection layer. The sender and the averaged recipient can be thought of as two global features acting as a shared condition of the environment when surveying the entire content of an email. Intuitively, the word embeddings capture the senders and the recipients when they are learned from email content.

More formally, every output word w_o is obtained by a softmax to maximize

$$p(w_o|w_i, ..., w_{i+n}, s, r_1, ..., r_m) = \frac{e^{y_{w_t}}}{\sum_{w \in V} e^{y_w}} \quad (1)$$

where s is a sender and r_1, ..., r_m represent m recipients. y_w refers to unnormalized log-probability for a word w in vocabulary V by

$$y = Xh(w_i, ..., w_{i+n};\ W, s, r_1, ..., r_m; U) + b \quad (2)$$

where X, b are the softmax parameters. W and U are matrix of word and user embeddings where $w_i, ..., w_{i+n}$ and $s, r_1, ..., r_m$ are extracted from. h is constructed by concatenating word and user embeddings in the order shown in Figure 1, defined as

$$h = v_s \oplus \sum_{j=i}^{i+n} v_j \oplus \frac{1}{m} \sum_{r=1}^{m} v_r \quad (3)$$

where v_s, v_j and v_r are embeddings of the sender, content words and recipients, respectively. Particularly, embeddings from input words are summed dimension-wise to the project layer, just like in the CBOW structure. Averaging over the embeddings of recipients in h is because we treat all recipients equally important and thus so are their contributions to the projection layer. For efficiency, we follow the hierarchical softmax optimization used in `word2vec` (Mikolov et al., 2013).

In general, this framework can be considered a step-by-step learner that traverses a user network derived from email headers (senders and recipients), where in each step the learner learns a partial network from one user node to others via edges of email communications. We note that, like conventional word embedding learning, our approach can be considered as an offline learner since the learned embeddings cannot represent users absent in training data. To address this, one can always introduce a special token to present unknown users in the training stage, which is a commonly adopted technique in word embedding learning.

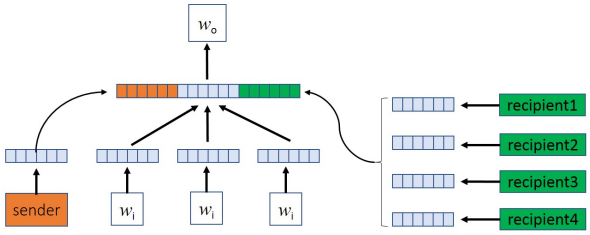

Figure 1: Our framework of learning user embeddings. Sender and recipients are mapped into corresponding embeddings and concatenated with the sum of word embeddings in the project layer. w_i and w_o refer to input and output words in email content.

User	Set1		Set2	
	#Folder	#Msg	#Folder	#Msg
beck-s	102	1795	78	1749
farmer-d	28	3677	25	3672
kaminski-v	37	2691	32	2684
lokay-m	12	2494	11	2493
sanders-r	31	1184	29	1181
williams-w3	20	2771	17	2766

Table 1: Email statistics for a selected set of users in Enron. Set1 removes non-topical folders while Set2 additionally disregards small folders.

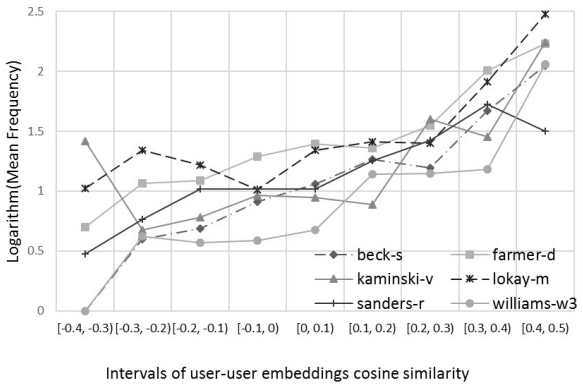

Figure 2: The similarity between users' embeddings positively correlates with the frequency that the two users communicates. X-axis: bucketed intervals of cosine similarity between users' embeddings. Y-axis: logarithm of average number of times emails being exchanged.

In a recent work proposed by (Yu et al., 2016), they obtained user embeddings through learning word embeddings from social texts. Their idea is similar to ours in terms of using a joint learning framework, but differs in two aspects. Their model relied on document vectors when trained directly or indirectly with word embeddings, while our framework does not require separate document embeddings in training. Furthermore, their user embeddings were averaged with word embeddings for next word prediction, which thus can be seen as a special type of word embeddings. In our approach user embeddings are concatenated with word embeddings in the projection layer, so that it can provide more explicit information when learning word embeddings.

Our work is also related to studies of learning vertex representation in social network (Perozzi et al., 2014; Tang et al., 2015; Cao et al., 2015). To represent user nodes, this line of work focused on analyzing network structure which is often formed by semantic edges (e.g., edges that indicate friendship or authorship). On the contrary, emails connect users in our work, meaning that the edges are composed of lexical content which provides more fine-grained signals than simple relational edges. This critical difference motivates us to design our framework, since the way prior methods connect users may result in a large number of isolated islands in email corpus, due to its lower degree of connectivity. Instead, our method represents users via learning the similar content they send/receive, which thus helps creating soft connections between users as long as "they speak the same language".

3 Experiments

We evaluate our approach using a publicly available email corpus, the Enron dataset (Klimt and Yang, 2004)[1]. The entire collection is considered for training user and word embeddings. We preprocess the documents using our in-house normalizers, which replace all URLs, Date, Time, Address, Phone Numbers with unified symbols, so as to reduce the sparsity of the data. The dimension of embeddings is set to 100.

Previous work on the auto-foldering task mainly focused on modeling message content and metadata to group together emails by their semantics. Bekkerman et al. (2004) extracted bag-of-words as document representation, whereas Dredze et al. (2008) adopted LDA to generate summary keywords for auto-foldering and recipient prediction. In recent work by Tam et al. (2012), multiple features were generated from different fields such as subject, body and participants. Grbovic et al. (2014) tackled email classification from a different angle. In their setup, the target folders were aggregated and inferred by running LDA on the entire corpus, which is different from the work that concentrates on predicting user defined folders.

To evaluate applying user embeddings to auto-foldering, we follow conventional settings (Bekkerman et al., 2004; Dredze et al., 2008; Tam et al., 2012) where personal emails from a set of users are adopted for prediction. Similar to previous work, we remove non-topical folders such as *Inbox*, *Sent-Items*, *Deleted Items*, etc., from the data, and further folders with a small number of messages, i.e., ≤ 3, are disregarded. The statistics

[1] http://www.cs.cmu.edu/~enron/

Learner	Approach	*beck-s*	*farmer-d*	*kaminski-v*	*lokay-m*	*sanders-r*	*williams-w3*	Avg
LR	SB	0.68	0.79	0.79	0.83	0.77	0.93	0.80
	SB+Emb	0.73	0.81	0.80	**0.87**	0.80	0.95	0.83
	SBFT	0.73	**0.82**	**0.81**	**0.87**	**0.82**	0.95	0.83
	SBFT+Emb	**0.74**	**0.82**	**0.81**	**0.87**	**0.82**	**0.96**	**0.84**
AP	SB	0.52	0.77	0.73	0.80	0.68	0.91	0.74
	SB+Emb	0.57	0.79	0.75	0.83	0.70	0.92	0.76
	SBFT	0.60	**0.80**	**0.76**	0.84	0.74	0.93	0.78
	SBFT+Emb	**0.61**	**0.80**	**0.76**	**0.85**	**0.75**	**0.94**	**0.79**
SVM	SB	0.53	0.76	0.72	0.79	0.65	0.92	0.73
	SB+Emb	0.57	0.78	0.73	0.83	0.68	0.93	0.75
	SBFT	0.59	0.78	0.76	0.84	0.72	**0.94**	0.77
	SBFT+Emb	**0.61**	**0.80**	**0.77**	**0.85**	**0.76**	**0.94**	**0.79**

Table 2: Accuracy results of classification methods on Set1 for selected Enron users. Highest accuracy for each user is marked bold for a given learner.

of these two subsets are shown in Table 1. We note that this Enron data set of version May 7, 2015 incorporates additional changes. Hence, compared to reports of prior work (Bekkerman et al., 2004; Tam et al., 2012), statistics in Table 1 show certain differences[2] and the absolute evaluation numbers are not directly comparable with theirs.

Our experiments are conducted using several popular classifiers: logistic regression (LR), averaged perception (AP), and support vector machine (SVM) to predict the most likely target folders. According to Dredze et al. (2008), the highest accuracy is achieved when the entire message is used in offline prediction. Tam et al. (2012) reported that the best performing results take into account the content of subject, body and participants. We reference the two findings as our baseline approaches: the first method featurizes each message with the n-grams of subject (S) and body (B), $n \in \{1, 2, 3\}$, whereas the second method further adds n-grams of the from (F) and to (T) fields in metadata. Our proposed approach, SB+Emb and SBFT+Emb, represents each email using a combination of lexical n-grams from SB(FT) and user embeddings (Emb) trained with the entire corpus.

3.1 User Embeddings Analysis

To understand if the learned user embeddings reflect actual email correspondence, we study the relation between the similarity of users' embeddings and the frequency they communicate. Specifically, for each target user u_i, we first identify all others $\{u_j|_{j\neq i}\}$ that he/she has had communications with, and then bucket the cosine similarity between their embeddings into intervals. For each interval, we take the average of the numbers of times each u_j communicates with u_i and convert it into logarithm space. Figure 2 shows that in general similarity between user embeddings positively correlates with the frequency those users send/receive emails to/from others. This implies the learned embeddings can capture users' interactions through words, therefore forming a fair user representation candidate.

We conduct the same analysis on user-word relation additionally. The results resembles previous observation that a word is more similar to a user (i.e., higher cosine score) if the word appears more often in the user's emails. Yet when a word becomes very frequent, it functions like a stopword thereby making this property no longer hold.

3.2 Auto-Foldering

Table 2 shows the overall accuracy results on data Set1. Across all learners and users, we observe a consistent pattern that SB+Emb improves the performance of SB with a varying percentage from 1% to 10%. This suggests that adding user embeddings provides extra signals regarding how users may organize information.

Comparing SB and SBFT, it is clear that taking into account participants is highly helpful for prediction, as indicated by Tam et al. (2012). The performance of SB+Emb is either comparable with or worse than SBFT. We think this may be because using n-grams of email addresses conveys more precise information regarding who were involved in an email communication, whereas embeddings operate on a denser semantic space without giving exact representation. Although SB+Emb may show some performance inferiority compared to SBFT, it provides much higher flexibility than ex-

[2]E.g., we omit data for the user *kitchen-l*, for the reason that it contains only 2 folders after preprocessing.

Learner	Approach	*beck-s*	*farmer-d*	*kaminski-v*	*lokay-m*	*sanders-r*	*williams-w3*	Avg
LR	SB	0.69	0.79	0.79	0.83	0.77	0.93	0.80
	SB+Emb	0.74	0.81	0.79	0.87	0.80	0.95	0.83
	SBFT	0.75	**0.82**	**0.81**	0.87	**0.83**	0.95	**0.84**
	SBFT+Emb	**0.76**	**0.82**	**0.81**	**0.88**	**0.83**	**0.96**	**0.84**
AP	SB	0.52	0.77	0.72	0.80	0.67	0.92	0.73
	SB+Emb	0.59	0.79	0.74	0.83	0.70	0.93	0.76
	SBFT	0.59	0.80	**0.76**	0.85	0.73	**0.94**	0.78
	SBFT+Emb	**0.62**	**0.81**	**0.76**	**0.86**	**0.76**	**0.94**	**0.79**
SVM	SB	0.52	0.77	0.73	0.79	0.66	0.92	0.73
	SB+Emb	0.58	0.78	**0.75**	0.83	0.69	0.93	0.76
	SBFT	0.61	0.79	0.74	0.83	**0.73**	0.93	0.77
	SBFT+Emb	**0.62**	**0.80**	**0.75**	**0.84**	**0.73**	**0.94**	**0.78**

Table 3: Accuracy results of classification methods on Set2 for selected Enron users. Highest accuracy for each user is marked bold for a given learner.

act matching and can better address properties for unseen or infrequent users. Therefore it could be the case that SB+Emb performs better categorization for larger audience in practice. When incorporating user embeddings on the top of all available lexical features (i.e., SBFT+Emb), prediction accuracy can be further increased compared to pure SBFT.

At an individual level, *beck-s* and *sanders-r* gain relatively the most when including user embeddings. Although these two users, especially *beck-s*, have more folders than others and thus present more challenges for classifiers, user embeddings has potential to effectively introduce user-token interactions for organizing information. On the contrary, the improvements based on embedding features are less apparent for *williams-w3*, whose folder categorization was the most unbalanced among all (i.e., a majority of emails belong to the same folder, making the prediction fairly easy with just few signals). Comparing different learners, we see that LR works the best in general, with AP and SVM performing somewhat comparable.

We conduct the same experiments on data Set2, which removes both non-topical and small folders. Table 3 shows that the overall trend is similar to what is observed in Table 2.

4 Conclusions and Future Work

In this paper, we proposed an approach to learning user embeddings from emails based on the sender-recipient network. Our analysis suggested that the learned embeddings reflect the interactions in the original corpus, where frequent emails exchangers tend to be more similar to each other. Evaluating from an application point of view, we showed that applying user embeddings to the auto-foldering task resulted in improved accuracy.

Yet another advantage of our approach is it learns meta-data in an unsupervised manner. As email data is highly private and sensitive, eyes-off techniques like ours not only bypass the need of human annotations but also leverage the information collected from the entire data. More importantly, using representations avoids leaking sensitive information delivered by lexical terms.

One direct follow-up of this work is learning user embeddings from social networks, or taking social network features into account. Learning task-specific embeddings is another direction to investigate as we move forward, e.g., modeling user-folder-words interactions for auto-foldering task with embeddings. Other tasks such as using embeddings for knowledge mining from emails, or online embedding training and updating with accumulating email data, will be interesting to explore. Finally, it will be important for us to test on larger, more realistic email datasets in the future.

References

Ron Bekkerman, Andrew Mccallum, and Gary Huang. 2004. Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora. In *In Technical Report, Computer Science Department, University of Massachusetts, IR-418*, pages 1–23.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. GraRep: Learning Graph Representations with Global Structural Information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 891–900, New York, NY, USA. ACM.

Giuseppe Carenini, Raymond T. Ng, and Xiaodong Zhou. 2008. Summarizing Emails with Conversa-

tional Cohesion and Subjectivity. In *Proceedings of ACL-08: HLT*, pages 353–361, Columbus, Ohio, June. Association for Computational Linguistics.

Mark Dredze, Hanna M. Wallach, Danny Puller, and Fernando Pereira. 2008. Generating Summary Keywords for Emails Using Topics. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, IUI '08, pages 199–206, New York, NY, USA. ACM.

H. Drucker, Donghui Wu, and V. N. Vapnik. 1999. Support Vector Machines for Spam Categorization. *Transaction on Neural Networks*, 10(5):1048–1054, September.

Kevin R. Gee. 2003. Using Latent Semantic Indexing to Filter Spam. In *Proceedings of the 2003 ACM Symposium on Applied Computing*, SAC '03, pages 460–464.

Mihajlo Grbovic, Guy Halawi, Zohar Karnin, and Yoelle Maarek. 2014. How Many Folders Do You Really Need?: Classifying Email into a Handful of Categories. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, CIKM '14, pages 869–878, New York, NY, USA. ACM.

Aditya Grover and Jure Leskovec. 2016. Node2Vec: Scalable Feature Learning for Networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864.

Bryan Klimt and Yiming Yang. 2004. The Enron Corpus: A New Dataset for Email Classification Research. In *ECML*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226. Springer.

Nicholas Kushmerick and Tessa Lau. 2005. Automated Email Activity Management: An Unsupervised Learning Approach. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, IUI '05, pages 67–74, New York, NY, USA. ACM.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

David D. Lewis and Kimberly A. Knowles. 1997. Threading Electronic Mail: A Preliminary Study. *Information Processing and Management*, 33(2):209–217, March.

Dawen Liang, Laurent Charlin, James McInerney, and David M. Blei. 2016. Modeling User Exposure in Recommendation. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 951–961.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint*, abs/1301.3781.

Einat Minkov, Ramnath Balasubramanyan, and William W. Cohen. 2008. Activity-centred Search in Email. In *CEAS*.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online Learning of Social Representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA. ACM.

Tony Tam, Artur Ferreira, and André Lourenço. 2012. Automatic Foldering of Email Messages: A Combination Approach. In *Proceedings of the 34th European Conference on Advances in Information Retrieval*, ECIR'12, pages 232–243, Berlin, Heidelberg. Springer-Verlag.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1067–1077, Republic and Canton of Geneva, Switzerland.

Yang Yu, Xiaojun Wan, and Xinjie Zhou. 2016. User Embedding for Scholarly Microblog Recommendation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 449–453, Berlin, Germany, August. Association for Computational Linguistics.

Bin Zhao, Weining Qian, and Aoying Zhou. 2010. Towards Bipartite Graph Data Management. In *Proceedings of the Second International Workshop on Cloud Data Management*, CloudDB '10, pages 55–62.