Privacy-Aware Personalized Entity Representations for Improved User Understanding

Levi Melnick Hussein Elmessilhy Vassilis Polychronopoulos Gilsinia Lopez
Yuancheng Tu Omar Zia Khan Ye-Yi Wang Chris Quirk
Microsoft
{lemeln,huahme,vapolych,gilopez,yuantu,omkhan,yeyiwang,chrisq}@microsoft.com

ABSTRACT

Representation learning has transformed the field of machine learning. Advances like ImageNet, word2vec, and BERT demonstrate the power of pre-trained representations to accelerate model training. The effectiveness of these techniques derives from their ability to represent words, sentences, and images in context. Other entity types, such as people and topics, are crucial sources of context in enterprise use-cases, including organization, recommendation, and discovery of vast streams of information. But learning representations for these entities from private data aggregated across user shards carries the risk of privacy breaches. Personalizing representations by conditioning them on a single user's content eliminates privacy risks while providing a rich source of context that can change the interpretation of words, people, documents, groups, and other entities commonly encountered in workplace data. In this paper, we explore methods that embed user-conditioned representations of people, key phrases, and emails into a shared vector space based on an individual user's emails. We evaluate these representations on a suite of representative communication inference tasks using both a public email repository and live user data from an enterprise. We demonstrate that our privacy-preserving light-weight unsupervised representations rival supervised approaches. When used to augment supervised approaches, these representations are competitive with deep-learned multi-task models based on pre-trained representations.

1 INTRODUCTION

Pre-trained embeddings are a crucial technique in machine learning applications, especially when task-specific training data is scarce. For instance, groundbreaking work in image captioning was enabled by reusing the penultimate layer of an object recognition system to summarize the content of an image[24]. More recently, contextualized embeddings are setting the state-of-the-art in a range of natural language processing tasks [12]. Training models to extract reusable representations from data is now an obvious investment. The next key research question is which context to leverage.

Our research is situated in the area of *User Understanding*: organizing the information, documents, and communications that are available to each user within an organization. Users now commonly retain huge mailboxes of written communication; members of larger organizations also have access to large repositories of access-controlled documents that are not publicly available. Our goals are to help each user find, classify, and act upon these growing information stores, then to acquire and organize information, including facts and relationships among these entities. A crucial enabling step is to build reusable representations of this information.

Most representation learning uses large, publicly-available document stores to build generic embeddings. We believe there is also great value in *user-conditioned representations*: representations of phrases and contacts for each user learned on the information uniquely available to that user. First, building user-conditioned representations provides a huge amount of context. Often when there are ambiguous or overloaded concepts, the key people surrounding their usage can disambiguate. Furthermore, a given user may extend the meanings of a given concept as they document and communicate new ideas. Perhaps most importantly, training a model based on only the communications and documents available to a given user provides a clear and intuitive notion of privacy. Whenever we train on data beyond any user's normal visibility, there is some potential for capturing and surfacing information outside their view. Differential Privacy helps limit the exposure of any individual user, but preventing leakage across groups is more difficult. For instance, certain privileged information may be discussed heavily by many members of an administrative board, yet this information should not be shared broadly across the whole organization. When training a user's model on only data that that user can see, the possibility for leaking information is removed. From the perspectives of both leveraging a crucial signal as well as maintaining user privacy and trust, user-conditioned representations hold great promise.

User-conditioned learning comes at a cost. Data density decreases dramatically. State-of-the-art deep learned representations typically train on billions of tokens [12], whereas an individual user's inbox may only have a few thousand emails. Thus, we explore shallower personalized approaches with lower sample complexity (though shallow models can be mixed with deep generic models for empirical gains [10]). Furthermore, training must be performed for every user separately within the organization; in our case, this entails separate training runs for hundreds of millions of users. Because the information available to the user is constantly changing, maintaining fresh representations is also a challenge.

As computation and storage become cheaper, the overhead of maintaining user-conditioned models is tractable only if the models are light-weight. Furthermore, we focus on task-agnostic representations that benefit a range of scenarios, amortizing the cost of computation. Finally, using models trained only on one user's data benefits privacy, which is an increasing concern for organizations and individuals.

PrivateNLP 2020, Feb 7, 2020, Houston, Texas

1.1 Contributions

We present the first efforts in building per-user representations: content-based representations that embed disparate entities, including contacts and key phrases, into a common vector space. These entity representations are different for each user: the same key phrase and contact may have very different representations across two users depending on their context. We focus on slow-changing entities like contacts and key phrases to minimize the impact of delayed retraining: although one's impression of their collaborators may shift over years, months, or perhaps weeks, a representation that is a few days old is still useful. To embed rapidly arriving and changing items such as documents and emails, we present approaches that assemble representations of rapidly changing entities from their content, including related contacts and key phrases.

We evaluate these representations on a range of downstream tasks, including action-prediction and content-prediction. Simple, unsupervised approaches, especially non-negative matrix factorization (NMF) [25], produce substantial improvements in accuracy, outperforming task-specific and multi-task neural network approaches.

We compare user-conditioned representations to representations learned at the organization-level, where data from multiple users are combined together into a larger undifferentiated store. User-conditioned representations mostly outperform the organization-level approaches, despite decreased data density, presumably because the additional context provides helpful signals to models. Furthermore, user-conditioned representations sidestep issues related to privacy preservation by not mixing data across user mailboxes.

2 RELATED WORK

Email mining [35] has been widely studied from different angles both for content and action classification. Spam detection has received considerable attention both from a content identification and filtering view [8] as well as from a process perspective [16]. Folder prediction is another task that can help better organize incoming emails [22]. Email content has also been used for social graph analysis to learn associations between people, both for recipient prediction [4] and sender prediction [17]. Action prediction tasks that have been considered in the context of emails include reply prediction [40], attachment prediction [15, 37], and generic email action prediction [6]. In this paper, we use recipient prediction, sender prediction, and reply prediction as representative tasks to evaluate the quality of our learned representations. All prior work on these and similar tasks has relied on per-task feature engineering and supervised model training. We show how entity representations generated in a task-agnostic manner can be used both in an unsupervised and in a supervised setting for these tasks.

Entity representations have been used extensively to provide personalized recommendations. Most such models build global representations of users and items in the same latent space and then determine the similarity between the user and item through cosine similarity. The user embeddings can be built in a collaborative filtering setting by leveraging a user's past actions such as clicks [30], structured attributes items interacted with, ratings offered on past items [5], or even past search queries [2]. An extension to such approaches is to combine embeddings of words or phrases with other types of data [33], such as embeddings of users [27] or their previously favored items [1]. Our entity representations also embed phrases with contacts, but they are task-agnostic.

Personalized language models have shown benefits in speech recognition [26], dialog [23], information retrieval [38], and collaborative filtering [19]. These approaches model the user, but the representations are not generated on a per-user level; instead, all users share the same representation for an item or phrase. In our approach, the same item (phrase or contact) will have a different representation according to each user, since the entity representations are generated per user only considering the data available to that user. Nawaz et al. [29] present a technique to perform social network analysis to identify similar communities of contacts within a user's own data but their approach is limited to a single task. Yoo et al. [41] describe an approach for obtaining representations of emails using personalized network connectivity features and contacts. Their representations are used as inputs to machine learning models to predict the importance of emails. In our approach, key phrases, contacts, and emails are all embedded in the same space. Thus, we can use the task-agnostic representations for a variety of tasks and obtain similarity scores for any pair of entities.

Starspace [39] introduces a neural-based embedding method that maps different types of entities (graphs, words, sentences, documents, etc.) to the same space. While the entities are embedded in the same space, like our work, their training uses global information, and the loss function on which the network is trained is task-dependent. Our approach allows reusing the same representations obtained using local data across a variety of tasks.

Our evaluation tasks bear some similarity to the evaluation of knowledge base completion through embeddings of text and entities [36]. However, our entities are not curated knowledge base entities; they are phrases and people known to a particular user.

In query expansion, locally trained embeddings can outperform global or pre-trained embeddings [13, 32] by incorporating more relevant local context. We exploit a similar insight in training only on the user's own data and directly incorporating her context. Amer et al. [3] present an approach that trains a per-user representation, though their per-user embeddings perform worse than global or pre-trained embeddings. In this paper, we demonstrate methods to train per-user representations that can not only outperform global representations but also generate them in a task-agnostic manner.

3 ENTITY REPRESENTATIONS

We developed representations for three entity types: key phrases, contacts, and emails. Key phrases (typically noun phrases) [18] can appear anywhere in the body or subject of an email. Restricting to extracted key phrases limits the total number of entities for which representations must be learned. By "contacts," we refer to individual email addresses that appear in the `From`, `To`, or `CC` fields of an email. For slow-changing entities like key phrases and contacts, we can periodically regenerate a stored representation. We represent fast-changing entities such as emails with light-weight compositions of pre-trained entity embeddings (contact and key phrase embeddings) to minimize computational expense.

Since the median inbox size will be small (in both our data sets it is around seven thousand emails) there is not enough data per user

to train useful deep learned representations. Indeed, our early attempts to train user-conditioned word2vec [28] embeddings yielded poor results and are not reported here. Therefore we only considered approaches that were likely to perform well at low data density.

3.1 Key Phrase and Contact Representations

We compute unsupervised entity representations for contacts and key phrases by associating them with *concatenated documents*. These concatenated documents are assembled from the user's original documents; the emails in our experiments have `From`, `To`, `CC`, `Body`, and `Subject` fields. Given a particular entity e, its concatenated document d_e is the concatenation of every email m in a user's inbox such that e appears in any of those fields. This concatenation is done on each field f independently: every new concatenated document d_e will have a corresponding field $d_{e,f}$ for each field f in the original document. We use $\bigoplus$ to denote concatenation.

$$d_{e,f} = \bigoplus_{m \in M_e} m_f, \text{ where } M_e = \{m : \exists f \text{ such that } e \in m_f\} \quad (1)$$

Stop words on the Scikit-learn stopword list are removed, as are terms that appear in more than 30% or less than 0.25% of training emails. We then generate a sparse numerical entity-by-term matrix, using TF-IDF for most methods or just TF matrix in the case of LDA. Initially one matrix is computed for each field f using the relevant portion of the concatenated document collection $D_f = \{d_{e,f}\}_e$. Each matrix is scaled according to a weighting factor w_f to balance its contributions, and finally these matrices are concatenated to form a single matrix T.

$$T = \bigoplus_f \sqrt{w_f} \cdot \text{term-matrix}(D_f), \text{ where } \sum_f w_f = 1 \quad (2)$$

The weights of the different email fields are treated as hyperparameters and tuned empirically to perform well on the evaluation tasks. We found that weights of 0.4, 0.3, 0.2, 0.05, and 0.05 for the `Body`, `Subject`, `From`, `To`, and `CC` fields worked well. The rows of T are sparse representations of entities – a simple and safe baseline. We explored LDA, LSA, and NMF as a means of encouraging softer matching through dimensionality reduction.

3.1.1 TF-IDF. Our baseline representation technique is sparse unigram TF-IDF vectors produced from the concatenated documents.

3.1.2 Latent Dirichlet Allocation (LDA). Latent topic models using LDA [7] over the term frequency matrix of the concatenated documents (not the TF-IDF matrix) learn a mixture of topics for each document. These learned vectors can act as entity representations. We can vary the number of latent topics to determine the dimensionality of the resulting embeddings.

3.1.3 Latent Semantic Analysis (LSA). A classic method for reducing sparsity, LSA [11] builds a low-rank approximation of a TF-IDF matrix T using the singular value decomposition: $T = U\Sigma V^T$.

3.1.4 Non-negative Matrix Factorization (NMF). The SVD reconstruction has a few problems: the values of the matrix may be positive or negative, and there is no explicit regularization term. Together, these issues may lead to strange or divergent weights, especially when the data is difficult to model with lower rank. Non-negative matrix factorization (NMF) addresses these issues by

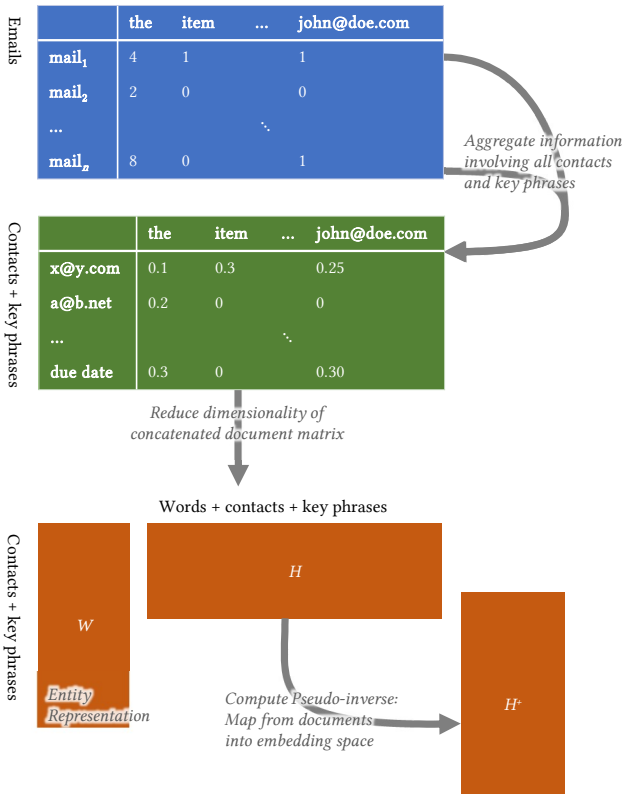

Figure 1: Process of creating concatenated documents to represent contacts and key phrases given the count matrix from an email corpus. We also demonstrate how this matrix can be factorized into a low rank approximation to encourage inference over sparse items. The left matrix W can be interpreted as entity representations. Furthermore, we can derive a mapping from the words, phrases, and contacts in an email into this representation space using the pseudoinverse H^+ of the right matrix H. Other low rank approximation and composition approaches are explored as well.

constraining the low-rank matrices to be positive and adding a regularization term [25]. Specifically, given an input matrix $T \in \mathbb{R}^{m \times n}$, we try to find matrices $W \in \mathbb{R}^{m \times d}$ and $H \in \mathbb{R}^{d \times n}$ to minimize

$$|T - WH| + \lambda (|W| + |H|) \quad (3)$$

where λ is a regularization weight and $|\cdot|$ is the Frobenius norm. The W matrix serves as a representation for the entities. We efficiently compute NMF through the Hierarchical Least Squares algorithm [21].

3.2 Email Representations

The vocabulary of key phrases and contacts in one's mailbox is likely to grow slowly, and their meanings and relationships will also evolve gradually. By comparison, many new emails arrive every day, so the "vocabulary" of email entities is constantly increasing. Thus, while it is possible to train representations for email in the same way that we do for key phrases and contacts, updating email representations on an ongoing basis would imply vast storage and

From: terrie.james@enron.com
Subject: Enron Prize
To: kenneth.lay@enron.com
Cc: cindy.olson@enron.com, kelly.kimberly@enron.com, karen.denne@enron.com, christie.patrick@enron.com, rosalee.fleming@enron.com
Date: Mon, 12 Nov 2001 16:51:33 -0800 (PST)

Replied to: False

Ken,

I wanted to let you know that I spoke with W.O. King at the Baker Institute earlier today. Everything is ready for tomorrow's activities.

There will be a great deal of media coverage for the event. Many media outlets will be accessing the live video feed of Chairman Greenspan's speech. However, the media will not be able to interview any participants directly. Chairman Greenspan will only answer written questions submitted by the audience (including media), and those questions will be vetted twice. I will be receiving the media advisory that has been distributed and the most current media attendee list from the Baker Institute tomorrow. I will provide that to you as early as possible.

Figure 2: Example email with sender, recipients, key phrases, and replied-to annotations. Each task is constructed by obscuring a relevant entity, then reconstructing it given the remaining context.

computation requirements. So we handle emails differently, computing representations on demand through compositions of other entity representations. In this paper, we explored four different email composition models: Centroid, Pointwise Max, Pseudoinverse, and the combination of Centroid and Pseudoinverse.

3.2.1 Centroid. One simple email representation is the average of the representations of all key phrases and contacts in an email.

3.2.2 Pointwise Max. Another commonly used pooling operation is max – we retain the largest value along each dimension. This approach increases the sensitivity to strongly-weighted features in the underlying key phrase and contact representations.

3.2.3 Pseudoinverse. The H matrix from Equation 3 can serve as a map from the low-rank concept space into the word/entity space. Although H is not a square matrix and hence not invertible, the Moore-Penrose pseudoinverse of H, namely H^+, can act as a map from email content into the entity representation space. We multiply the TF-IDF vector associated with a given email by H^+ to project into the entity representation space. Unlike the previous two models, this has the benefit of including information from non-key phrase unigrams from the email.

3.2.4 Centroid + Pseudoinverse. Centroid and pseudoinverse representations are summed to combine the benefits of each.

4 EVALUATION METHODOLOGY

4.1 Evaluation Tasks

We evaluate entity representations according to their performance on four email mining tasks: sender prediction, recipient prediction, related key phrase prediction, and reply prediction. The first three tasks are content prediction tasks, whereas in reply prediction we use the email content to predict a user action.

Content prediction tasks are formulated as association tasks. We remove a target entity from an email and randomly select nineteen distractor entities from the user's inbox not already present in the email. We use the cosine similarity between the email representation and the twenty candidate target entity representations to predict the true target. Reply prediction is treated as a binary classification problem, using email representations as input features. Entity representation methods that yield more accurate predictions are considered superior.

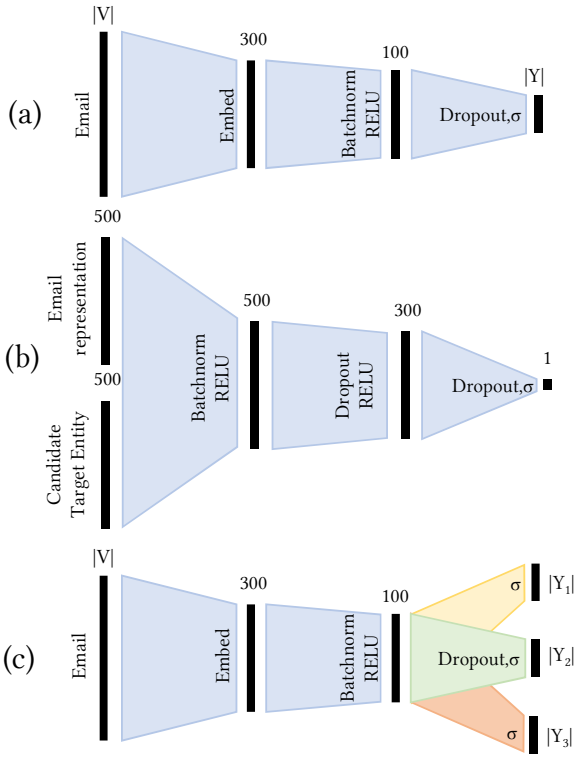

Figure 3: Task-specific neural network architectures: (a) multiclass model for predicting which entity is present in a given email; (b) binary matching model for predicting whether a given entity is present in a given email; (c) multi-task multiclass model jointly trained on all evaluation tasks.

These tasks readily suggest real life applications. Recipient recommendation is already a standard feature in many email clients. Similarly, an email client may predict whether an email from an unfamiliar address comes from a known sender and prompt the user to add the new address to that sender's contact information. Predicting latent associations between emails and key phrases enables automatic topic tagging and foldering. Finally, an email client may use reply prediction to identify important emails to which an inbox owner has not yet responded and remind the user to reply.

4.1.1 Task-Specific Model Architectures. We aim to construct task-agnostic user-conditioned representations: they should be useful across a variety of tasks without having to be tuned to each one separately. While this makes the representations reusable and reduces computational expense, separate models trained on each specific task often perform better. To evaluate this tradeoff, we compare the unsupervised similarity-based method described above to supervised task-specific baseline models trained on each of the

association tasks. We also evaluate how well the user-conditioned representations perform as feature inputs to task-specific models, since their utility as feature representations is a key consideration.

To train a task-specific model for sender, recipient, or key phrase prediction, we reformulate these association tasks as classification problems. In each case, we train the classifier to predict the target entity using its email representation. As above, we remove a target entity from an email and select nineteen distractors. Instead of cosine similarity, we use the trained classifier to score the twenty candidate entities and predict the one with the highest score.

We experimented with a variety of modeling techniques for both task-specific baseline models and task-specific models trained on entity representations. The best results consistently came from simple two-layer feed forward neural classifiers using ReLU activations, a sigmoid output layer, batch normalization, drop out [34], and trained using cross-entropy loss and Adam [14]. However, each scenario achieved best results using slightly different task formulations and architectures.

The baseline models were formulated as multiclass classifiers, as depicted in Figure 3a. Emails are represented as binary vectors with each element representing the presence or absence of a unigram or contact. These vectors index into a 300 dimensional embedding layer initialized with pre-trained GloVe vectors [31]; out-of-vocabulary items received random initializers. The embedding layer was also trained, allowing the model to learn representations for out-of-vocabulary terms. We experimented with two variants: one in which contacts were included as features ("Pre-trained + Contacts") and one in which they were not ("Pre-trained").

For models trained on entity representations, shown in Figure 3b, we found the best results by treating the candidate target entity representations and the email representations as separate inputs. These 500 dimensional representations are passed through two dense layers of width 500 and 300 respectively and a sigmoid output layer, which returns a score representing the likelihood that the input entity is indeed present in the input email. In Tables 4 and 5, "TF-IDF + NMF Centroid" and "TF-IDF + NMF Centroid + Pseudoinverse" model variants both share this architecture.

We also considered a multitask model jointly trained on all four evaluation tasks. This model, shown in Figure 3c, is identical in its architecture and training to the task-specific baseline model in Figure 3a except that instead of one output layer it has $|N|$ output layers corresponding to the $|N|$ tasks. Relative loss weights were used to balance the training impact from each task since the tasks had varying numbers of training examples.

4.2 Evaluation Metrics

We measure our performance on the association tasks through *accuracy* (percentage of successful predictions) and *average recall*. A successful prediction is one where the target entity is scored highest among all candidates. Since there is one target and nineteen distractors, random guessing achieves an accuracy of 0.05.

Accuracy can allow a small number of frequently occurring entities to have a disproportionate effect. For instance, in sender prediction the majority of emails may be from a small set of senders: performance on these senders will skew the results. Thus, we also report the average recall, an efficient measure for skewed distributions. To obtain the average recall, we calculate the recall for each possible target: the percentage of times it was successfully predicted. We then report the average recall over all targets without weighing the frequency of the target. Together, accuracy and average recall provide a reliable measure of the association. If one method boosts accuracy by only learning about frequent targets, the average recall will be impacted negatively. Similarly a reduced recall of the frequent targets will impact the accuracy.

For reply prediction, we report the area under the precision-recall curve (PR-AUC), which is useful even when classes are imbalanced.

	Avocado (55 users)			Enterprise (53 users)		
	Max	Min	Average	Max	Min	Average
Emails/User	19,000	3,561	7,887	17,490	2,872	8,451
Phrases/User	9,632	3,324	5,308	8,137	3,433	6,772
Contacts/User	376	95	210	2,375	357	1,431
Reply Rate	0.34	0.01	0.14	0.60	0.01	0.19

Table 1: Email statistics for Avocado and enterprise users.

4.3 Evaluation Corpora

We evaluate our techniques on two separate repositories of emails, Avocado emails and live user emails from a large enterprise. The properties for each corpus are listed in Table 1. For the first repository, we use mailboxes from the Avocado Research Email Collection[1]. For the second dataset, we use live user email data from a real-world enterprise with thousands of users (called enterprise users from here on for brevity). These emails are encrypted and off-limits to human inspection. We randomly select a set of users who are related to each other by sampling from the same department. This increases the possibility of overlap between users and allows some shared context. This property will be helpful when we want to compare a global model versus user-conditioned representations.

For both datasets, we filter out users with fewer than 3,500 or greater than 20,000 emails. Users with more than 20,000 emails were outliers and, in the enterprise dataset, were likely to have many machine generated emails, which can make the evaluation tasks easier. We set the minimum number of emails to 3,500 somewhat arbitrarily because in our enterprise scenario it is almost always possible to obtain this many for a given user by extending the date range. We plan to investigate the performance of user-conditioned representations produced from smaller inboxes in future work.

5 EXPERIMENTS

We show that user-conditioned entity representations outperform strong global model baselines. NMF applied to our version of TF-IDF matrices proves most effective among the methods surveyed for representing key phrases and contacts. The combination of centroid and pseudoinverse methods detailed in Section 3.2.4 works best for composing email representations. While on some tasks supervised task-specific baseline models achieved higher accuracy than entity representation similarity-based methods, the latter were competitive and had significantly better recall. Task-specific models trained

[1] https://catalog.ldc.upenn.edu/LDC2015T03

Method	Sender		Recipient		Rel. Phrase	
	Acc	Rec	Acc	Rec	Acc	Rec
TF-IDF	0.59	0.28	0.59	0.31	0.60	0.41
LDA	0.53	0.37	0.51	0.41	0.49	0.42
LSA	0.59	0.29	0.59	0.32	0.60	0.42
NMF unreg.($\lambda = 0$)	0.61	0.37	0.59	0.40	0.60	0.46
NMF ($\lambda = 0.0001$)	**0.62**	**0.40**	**0.62**	**0.44**	**0.66**	**0.53**

Table 2: Evaluation task performance of key phrase and contact representation methods. In every case, the tasks use the Centroid method for composing email representations. Avocado data set.

Method	Sender		Recipient		Rel. Phrase	
	Acc	Rec	Acc	Rec	Acc	Rec
Centroid	0.62	0.40	**0.62**	0.44	**0.66**	0.53
Pointwise max	0.59	0.30	0.59	0.34	0.61	0.42
Pseudoinverse	0.49	**0.56**	0.47	**0.56**	0.55	**0.58**
Centroid+Pseudoinv	**0.64**	0.53	**0.62**	0.54	**0.66**	0.53

Table 3: Evaluation task performance of email representation methods. In every case, the tasks use regularized NMF to produce key phrase and contact representations. Avocado data set.

on entity representations also outperformed task-specific models trained on baseline features, demonstrating the entity representations' value as feature inputs. Our results here also show that entity representations are competitive with multitask learning despite the fact that they are trained without knowledge of the downstream tasks. We discuss these results in the following subsections.

5.1 User-Conditioned Representations

Slow Changing Entities: Key Phrases and Contacts. We compare unsupervised methods for producing key phrase and contact representations in Table 2. For LDA, LSA, and NMF, we perform hyperparameter tuning on a single enterprise user and report results for all techniques with their best settings. Since the evaluation tasks require representations for email as well as key phrases and contacts, we use the Centroid email representation in each case to ensure a fair comparison. Predictions are based on cosine similarity.[2] NMF with regularization outperformed all other methods. Regularization leads to more effective representations for NMF; comparing unregularized NMF to LSA suggests that non-negativity is also a helpful bias. Some of the most substantial gains are in recall, especially when compared to sparse TF-IDF baselines.

Composition for Fast Changing Entities: Email. Different compositional operations for representing email are explored in Table 3. Because NMF performed best across all tasks, we restrict our attention to these representations. The centroid method outperforms others on accuracy, though the pseudoinverse approach is the best for recall, presumably because it can incorporate information from unigrams in the represented email and not just the key phrases and contacts. A linear combination of centroid and pseudoinverse representations provides the best results for accuracy and almost matches pseudoinverse for recall.

5.2 Task-Specific Models

Task-Agnostic vs. Task-Specific. Unsupervised, task-agnostic approaches are versatile and reusable, but they may underperform relative to supervised models tuned to specific tasks. As described in Section 4.1.1, we explore this tradeoff by comparing the performance of entity representation similarity-based methods against task-specific baseline models trained on the evaluation tasks. For Avocado, we see that while the accuracy is indeed better on task-specific Pre-trained and Pre-trained + Contacts compared to the best representation methods (TF-IDF + NMF Centroid and TF-IDF NMF Centroid + Pseudoinverse), [3] as shown in Table 4. However, the TF-IDF + NMF Centroid + Pseudoinverse representations achieved significantly better recall for all three content prediction tasks and better accuracy in key phrase prediction, again indicating their ability to avoid over-optimizing for frequently occurring entities. This model produces even better results on the enterprise data set, where its accuracy is competitive with both of the Pre-trained models and its improvement in recall is even more dramatic. The higher number of contacts in the enterprise set enables better joint modeling with the content, allowing the entity representations to perform better in this setting. We can see that unsupervised entity representations are competitive with supervised baselines.

Entity Representations as Input Features. As our results suggest, user-conditioned entity representations are useful as input features to supervised models. To assess their value as feature representations, we compare task-specific models trained on entity representations with task-specific baselines, as described in Section 4.1.1. On Avocado, the entity representation-based task-specific models, TF-IDF + NMF Centroid and TF-IDF NMF Centroid + Pseudoinverse, outperform (or in a few cases match) the baselines on every task and metric. We see similar results on enterprise data, except a marginally lower reply prediction PR-AUC with entity-based task-specific models. Comparing the Avocado and enterprise results, we can see that the performance on all tasks is much better on enterprise users. Our hypothesis is that the larger contact vocabulary in enterprise (1,431 contacts per user on average) compared to Avocado (average 210 contacts per user) makes sender and recipient tasks easier: the distractors are sampled from a larger pool of contacts, and therefore less likely to be related and easier to screen out. In the case of reply prediction, we believe the higher PR-AUC stems from enterprise users that receive a higher volume of machine-generated emails, which have more predictable reply behavior.

5.3 User-Conditioned vs. Global Models

Each set of user-conditioned representations is trained on much fewer data than most representation learning techniques, but personalization is a powerful source of context. While our primary reason for focusing on user-conditioned entity representations is to avoid privacy leaks, we want to know how they compare against

[2] Reply prediction is difficult to evaluate in an unsupervised setting; hence, it is not reported here.

[3] Our results for sender and recipient prediction through an unsupervised task-agnostic representation are in the same range as those reported by Graus et al., [17] (0.66).

Data	Method	Sender		Recipient		Related Phrase		Reply
		Accuracy	Recall	Accuracy	Recall	Accuracy	Recall	PR-AUC
Avocado users	*Unsupervised Similarity-Based Methods*							
	TF-IDF + NMF Centroid	0.62	0.40	0.62	0.44	0.66	0.53	N/A
	TF-IDF + NMF Centroid + Pseudoinverse	0.64	**0.53**	0.62	**0.54**	**0.67**	**0.60**	N/A
	Supervised Task-Specific Models							
	Pre-trained	0.72	0.38	0.67	0.31	0.59	0.36	0.21
	Pre-trained + Contacts	0.74	0.42	0.71	0.35	0.60	0.37	0.24
	TF-IDF + NMF Centroid	0.74	0.48	0.72	0.47	0.64	0.49	0.28
	TF-IDF + NMF Centroid + Pseudoinverse	0.74	0.49	0.73	0.47	**0.67**	0.52	0.28
	Supervised Multi-Task Models							
	Pre-trained	0.73	0.47	0.69	0.42	0.59	0.35	0.28
	Pre-trained + Contacts	**0.78**	0.51	**0.75**	0.46	0.59	0.35	**0.30**
Enterprise users	*Unsupervised Similarity-Based Methods*							
	TF-IDF + NMF Centroid	0.81	0.73	0.86	0.79	0.69	0.60	N/A
	TF-IDF + NMF Centroid + Pseudoinverse	0.81	**0.77**	0.86	**0.81**	0.70	**0.65**	N/A
	Supervised Task-Specific Models							
	Pre-trained + Contacts	0.83	0.54	0.87	0.50	0.70	0.44	0.71
	TF-IDF + NMF Centroid	**0.87**	0.68	**0.91**	0.71	0.72	0.56	0.69
	TF-IDF + NMF Centroid + Pseudoinverse	**0.87**	0.70	**0.91**	0.72	**0.74**	0.59	0.65
	Supervised Multi-Task Models							
	Pre-trained + Contacts	0.85	0.58	0.88	0.54	0.70	0.43	**0.72**

Table 4: Task-specific models trained using representations as features, for both enterprise and Avocado users.

non-privacy-aware "global" representations trained on data from every user in an organization. In Table 5 we see that user-conditioned representations are significantly better on all tasks across all metrics compared to the global versions of those representations. This indicates that, for these models, the local context of a user is more important than training on a larger data set. We see a similar trend with the Pre-trained + Contacts and Global Pre-trained + Contacts models, though the global variant outperforms the user-conditioned one in sender prediction on Avocado. On reply prediction, global models trained using representations perform similarly to Yang et al. [40] without any task-specific feature engineering.

5.4 Unsupervised vs. Multi-Task Approaches

Our primary focus has been unsupervised entity representation computation. An alternative approach is to induce representations in a multitask learning setting [9]. Multitask models often achieve better performance than separate models trained on the same tasks and, indeed, as seen in Table 4, the multitask model described in Section 4.1.1 outperforms task specific models trained on the same Pre-trained + Contacts feature representation.

On Avocado, the best multitask model achieves significantly better accuracy in sender and recipient prediction than Pre-trained and Pre-trained + Contacts methods; entity-based task-specific methods are still competitive on recall. We observe the same trend with enterprise, where multi-task models outperform task-specific Pre-trained + Contacts, though entity-based task-specific models outperform multi-task on all tasks and metrics except reply prediction PR-AUC.

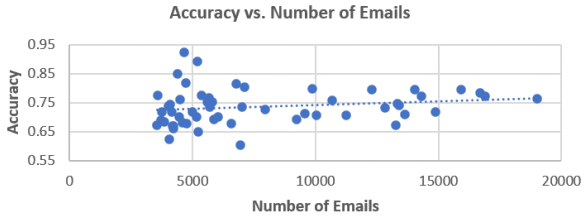

Figure 4: Sender prediction accuracy vs. number of training emails for TF-IDF + NMF Centroid on Avocado.

Thus the unsupervised methods presented here are competitive with multitask learning on recall despite the fact that they are trained without knowledge of the downstream tasks, and the task-specific entity-based models are competitive with the multi-task method on accuracy and better on recall.

5.5 The Effect of Data Size and Dimension

To explore the impact of data density, Figure 4 plots sender prediction accuracy using TF-IDF + NMF Centroid representations against the number of emails in a user's mailbox. Accuracy does not vary substantially, though average recall improves: additional data benefits representing entities on average. Similar trends for other tasks and other models were observed.

We plot the effect of varying dimension sizes for all tasks using the TF-IDF + NMF Centroid approach in Figure 5 for Avocado users. Representations of dimension 400 and 500 consistently achieve best results for both accuracy and recall.

Data	Method	Sender		Recipient		Related Phrase		Reply
		Accuracy	Recall	Accuracy	Recall	Accuracy	Recall	PR-AUC
Avocado users	*Unsupervised Similarity-Based Methods*							
	TF-IDF + NMF Centroid	0.62	0.40	0.62	0.44	0.66	0.53	N/A
	TF-IDF + NMF Centroid + Pseudoinverse	0.64	0.53	0.62	**0.54**	**0.67**	**0.60**	N/A
	Global TF-IDF + NMF	0.50	0.29	0.41	0.27	0.45	0.30	N/A
	Global TF-IDF + NMF Centroid + Pseudoinverse	0.55	0.40	0.40	0.34	0.43	0.37	N/A
	Supervised Task-Specific Models							
	Pre-trained + Contacts	0.74	0.42	0.71	0.35	0.60	0.37	0.24
	TF-IDF + NMF Centroid	0.74	0.48	0.72	0.47	0.64	0.49	**0.28**
	TF-IDF + NMF Centroid + Pseudoinverse	0.74	0.49	**0.73**	0.47	**0.67**	0.52	**0.28**
	Global Pre-trained + Contacts	**0.77**	**0.63**	0.65	0.48	0.58	0.34	0.21
	Global TF-IDF + NMF Centroid	0.70	0.50	0.58	0.36	0.52	0.29	0.25
	Global TF-IDF + NMF Centroid + Pseudoinverse	0.71	0.52	0.57	0.37	0.53	0.30	0.19
Enterprise users	*Unsupervised Similarity-Based Methods*							
	TF-IDF + NMF Centroid	0.81	0.73	0.86	0.79	0.69	0.60	N/A
	TF-IDF + NMF Centroid + Pseudoinverse	0.81	**0.77**	0.86	**0.81**	0.70	**0.65**	N/A
	Global TF-IDF + NMF	0.50	0.49	0.45	0.30	0.45	0.41	N/A
	Global TF-IDF + NMF Centroid + Pseudoinverse	0.48	0.51	0.43	0.34	0.45	0.43	N/A
	Supervised Task-Specific Models							
	Pre-trained + Contacts	0.83	0.54	0.87	0.50	0.70	0.44	**0.71**
	TF-IDF + NMF Centroid	**0.87**	0.68	**0.91**	0.71	0.72	0.56	0.69
	TF-IDF + NMF Centroid + Pseudoinverse	**0.87**	0.70	**0.91**	0.72	**0.74**	0.59	0.65
	Global Pretrained + Contacts	0.80	0.61	0.77	0.44	0.63	0.46	0.65
	Global TF-IDF + NMF Centroid	0.61	0.49	0.47	0.25	0.49	0.34	0.67
	Global TF-IDF + NMF Centroid + Pseudoinverse	0.60	0.51	0.48	0.30	0.50	0.36	0.56

Table 5: Individual vs. global models on Avocado and enterprise users.

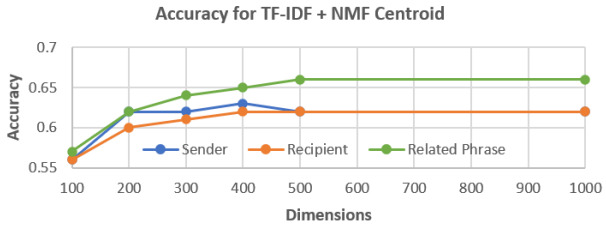

Figure 5: Effect of dimensionality on entity representations.

5.6 Practical Implications

Our current implementation has several optimizations intended for a production environment. We maintain updates to the TF-IDF matrix in a streaming manner upon receipt of each email. A periodic task, run every few days to every week, computes a fresh NMF representation, using approximately one minute of computation time per user with an optimized implementation based on Sparse BLAS operations in Intel MKL. [20] The process is running constantly for thousands of users, scaling up to hundreds of thousands of users.

6 CONCLUSIONS

We have demonstrated approaches for learning task-agnostic user-conditioned embeddings that outperform strong baselines and demonstrate value in a range of downstream tasks. User-conditioned approaches are privacy preserving and show substantial benefits over global models, despite their lower data density. These promising results suggest a range of future directions to explore. One clear next step is to extend our approach to include documents, meetings, and other enterprise entities. Beyond that, embedding relationships between entities could help in predicting more complex connections between them. Next, our explorations in multitask modeling suggest that generalization across tasks also has value. Evaluating the impact of multitask representations on new tasks through leave-one-out experiments may help quantify this.

REFERENCES

[1] Q. Ai, V. Azizi, X. Chen, and Y. Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9):137, 2018.
[2] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Croft. Learning a hierarchical embedding model for personalized product search. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 645–654, 2017.
[3] N. O. Amer, P. Mulhem, and M. Géry. Toward word embedding for personalized information retrieval. *CoRR*, abs/1606.06991, 2016.
[4] R. Balasubramanyan, V. R. Carvalho, and W. Cohen. Cutonce-recipient recommendation and leak detection in action. In *AAAI, Workshop on Enhanced Messaging*, 2008.
[5] Y. Bao, H. Fang, and J. Zhang. TopicMF: Simultaneously exploiting ratings and reviews for recommendation. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.
[6] P. N. Bennett and J. G. Carbonell. Combining probability-based rankers for action-item detection. In *Human Language Technologies: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of*

the Main Conference, pages 324–331, 2007.
[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
[8] A. Bratko, B. Filipič, G. V. Cormack, T. R. Lynam, and B. Zupan. Spam filtering using statistical data compression models. *JMLR*, 7:2673–2698, Dec. 2006.
[9] R. Caruana. Multitask learning. *Mach. Learn.*, 28(1):41–75, July 1997.
[10] M. X. Chen, B. N. Lee, G. Bansal, Y. Cao, S. Zhang, J. Lu, J. Tsay, Y. Wang, A. M. Dai, Z. Chen, T. Sohn, and Y. Wu. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pages 2287–2295, New York, NY, USA, 2019. ACM.
[11] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.
[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
[13] F. Diaz, B. Mitra, and N. Craswell. Query expansion with locally-trained word embeddings. *CoRR*, abs/1605.07891, 2016.
[14] J. B. Diederik Kingma. Adam: A method for stochastic optimization. In *arXiv:1412.6980*, 2014.
[15] M. Dredze, T. Brooks, J. Carroll, J. Magarick, J. Blitzer, and F. Pereira. Intelligent email: Reply and attachment prediction. In *Proceedings of the 13th International ACM Conference on Intelligent user interfaces*, pages 321–324, 2008.
[16] J. Goodman, G. V. Cormack, and D. Heckerman. Spam and the ongoing battle for the inbox. *Communications of the ACM*, 50(2):24–33, 2007.
[17] D. Graus, D. Van Dijk, M. Tsagkias, W. Weerkamp, and M. 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*, pages 1079–1082, 2014.
[18] K. S. Hasan and V. Ng. Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1262–1273, Baltimore, Maryland, June 2014. Association for Computational Linguistics.
[19] G. Hu. Personalized neural embeddings for collaborative filtering with text. *arXiv preprint arXiv:1903.07860*, 2019.
[20] Intel. *Intel Math Kernel Library. Reference Manual.* Intel Corporation, Santa Clara, USA, 2009. ISBN 630813-054US.
[21] J. Kim, Y. He, and H. Park. Algorithms for nonnegative matrix and tensor factorizations: a unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319, Feb 2014.
[22] B. Klimt and Y. Yang. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*. Springer, 2004.
[23] S. Kottur, X. Wang, and V. Carvalho. Exploring personalized neural conversational models. In *IJCAI*, pages 3728–3734, 2017.
[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
[25] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
[26] M. Levit, A. Stolcke, R. Subba, S. Parthasarathy, S. Chang, S. Xie, T. Anastasakos, and B. Dumoulin. Personalization of word-phrase-entity language models. In *Proc. Interspeech*, pages 448–452. ISCA - International Speech Communication Association, September 2015.
[27] S. Liang, X. Zhang, Z. Ren, and E. Kanoulas. Dynamic embeddings for user profiling in twitter. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1764–1773, 2018.
[28] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*.
[29] W. Nawaz, Y. Han, K.-U. Khan, and Y.-K. Lee. Personalized email community detection using collaborative similarity measure. *arXiv preprint:1306.1300*, 2013.
[30] T. Nguyen and A. Takasu. Npe: neural personalized embedding for collaborative filtering. *arXiv preprint arXiv:1805.06563*, 2018.
[31] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.
[32] A. Rattinger, J.-M. L. Goff, and C. Gütl. Local word embeddings for query expansion based on co-authorship and citations. In *BIR@ECIR*, 2018.
[33] M. Rudolph, F. Ruiz, S. Mandt, and D. Blei. Exponential family embeddings. In *Advances in Neural Information Processing Systems*, pages 478–486, 2016.
[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
[35] G. Tang, J. Pei, and W.-S. Luk. Email mining: tasks, common techniques, and tools. *Knowledge and Information Systems*, 41(1):1–31, 2014.
[36] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon. Representing text for joint embedding of text and knowledge bases. In *EMNLP*, 2015.
[37] C. Van Gysel, B. Mitra, M. Venanzi, R. Rosemarin, G. Kukla, P. Grudzien, and N. Cancedda. Reply with: Proactive recommendation of email attachments. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM, pages 327–336, New York, NY, USA, 2017.
[38] J. B. P. Vuurens, M. Larson, and A. P. de Vries. Exploring deep space: Learning personalized ranking in a semantic space. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, 2016.
[39] L. Y. Wu, A. Fisch, S. Chopra, K. Adams, A. Bordes, and J. Weston. Starspace: Embed all the things! In *AAAI*, 2018.
[40] L. Yang, S. T. Dumais, P. N. Bennett, and A. H. Awadallah. Characterizing and predicting enterprise email reply behavior. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 235–244, 2017.
[41] S. Yoo, Y. Yang, F. Lin, and I.-C. Moon. Mining social networks for personalized email prioritization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 967–976, 2009.