*Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 1336–1345
Marseille, 11–16 May 2020

# Email Classification Incorporating Social Networks and Thread Structure

**Sakahr Alkhereyf** *, **Owen Rambow** †
*Columbia University, † Elemental Cognition
New York, NY, USA
sakhar@cs.columbia.edu, owen.rambow@gmail.com

## Abstract

Existing methods for different document classification tasks in the context of social networks typically only capture the semantics of texts, while ignoring the users who exchange the text and the network they form. However, some work has shown that incorporating the social network information in addition to information from language is effective for various NLP applications including sentiment analysis, inferring user attributes, and predicting inter-personal relations. In this paper, we present an empirical study of email classification into "Business" and "Personal" categories. We represent the email communication using various graph structures. As features, we use both the textual information from the email content and social network information from the communication graphs. We also model the thread structure for emails. We focus on detecting personal emails, and we evaluate our methods on two corpora, only one of which we train on. The experimental results reveal that incorporating social network information improves over the performance of an approach based on textual information only. The results also show that considering the thread structure of emails improves the performance further. Furthermore, our approach improves over a state-of-the-art baseline which uses node embeddings based on both lexical and social network information.

**Keywords:** text classification, email classification, social networks, graph algorithms

## 1. Introduction

There has been much work on using social networks to predict user characteristics. This work exploits homophily (for example, young people are more likely to communicate with other young people). In contrast, there has been far less work that uses the communication network (the network induced by conversations) to improve document classification of the communications themselves. This is a harder problem, since homophily is not relevant when characterizing the communications themselves: in various document classification tasks, the document category might not be directly inferred from the relationship of the participants when the same participants exchange different types of documents. For instance, the same people might exchange both personal and business emails, or urgent and nonessential emails. In this paper, we study document classification in the context of written conversations. As our task, we choose classification of email into personal or business emails. There are several reasons for this choice.

1. We are interested in how personal relationships affect communication, taking into account that the same pair of people may have multiple types of relationships.

2. The task we choose is relevant. Email remains a crucial communication medium for both individuals and organizations for both personal and business communications. Kiritchenko and Matwin (2011) show that a typical user daily receives 40-50 emails. And despite the massive growth of other social media over the past decade, company email is still used for personal purposes as the recent Avocado corpus shows (section 3.).

3. Two large data sets are available, the Enron corpus and a data set of emails from an anonymous defunct information technology company referred to as Avocado.

4. Unlike other text classification tasks, particularly for emails (e.g. spam filtering), email classification into business and personal has not received much attention and it remains a challenging (as shown in the human inter-annotator agreement reported in (Alkhereyf and Rambow, 2017; Jabbari et al., 2006)) and unsolved task.

5. We are interested in how people communicate in conversations, and email has real conversations. This distinguishes email from blogs and Twitter, which are readily available, but typically used for broadcasting to a large group of followers.

As for any document classification task, the language used (reflecting both content and language style) is highly predictive of the class. For instance, when a student speaks with her friends, she will probably use relatively less formal language than when she speaks with her professor, and she will talk about different topics. As we will see, using word embeddings provides a strong baseline for our task.

In this paper, our task is to use the textual content of documents and the underlying social network of email exchange for email classification into two categories, "Business" and "Personal". We use two annotated e-mail datasets, Enron and Avocado. We model the task of finding the rarer class (personal emails) in a set of all emails. We are interested in developing models that can be applied to unseen datasets, so that we can detect personal emails in new datasets with no retraining.

The specific contributions of this paper are as follows:

- It is not obvious how to model the email communication as a social network for the classification task in this paper. We extract features of emails from various graph structures representing the email exchange network and then use these features with machine learning models.

- We show that that a combination of social network information and email content leads to classification improvements over the performance of an approach based on textual information only.
- We show that by adding sequential modeling of threads (conversations), we get an important improvement in performance, significantly outperforming the individual email modeling approach. We have thus established that modeling the social network helps in document classification, but modeling the thread structure is also important.
- We show that our approach outperforms a state-of-the-art method proposed in the literature based on node embeddings, namely GraphSAGE.

Because we are interested in modeling thread structure, we use datasets which maintain the integrity of the thread (i.e. all emails belong to threads and all threads have labeled emails) and which we introduced in our previous work (Alkhereyf and Rambow, 2017). The Enron dataset is based on Columbia's Enron release (Agarwal et al., 2012). This paper adds the following research to our previous publication:

- We use neural network models.
- We use a strong baseline based on graph embeddings, namely, GraphSAGE (sections 5. and 6.3.).
- We explicitly model email threads (subsection 6.4.).
- We use word embeddings trained on our data (section 4.).

Also, as part of the submission we release the annotated Enron corpus in addition to other annotations including power relations as a language resource (Agarwal et al., 2020). For Avocado, we release the annotation labels with their corresponding email ids without the email content (because of licensing restrictions on the corpus itself) (Alkhereyf and Rambow, 2020).

The paper is organized as follows: we first review related literature in section 2., and then describe our datasets in section 3.. We discuss lexical features in section 4.. We present our baseline, a state-of-the-art node embedding model, in section 5.. Then we show how we model emails as a social network in section 6.. We present the experimental study to evaluate our models in section 7., and conclude in section 8..

# 2. Related Work

## 2.1. Incorporating Network and Language Information

Many previous studies on various natural language processing tasks in the context of social networks mainly focus on textual information and ignore other information that can be extracted from the underlying social network. However, there are some studies that incorporate the social network structure to improve the performance for different tasks including: inferring user attributes (Filippova, 2012; Al Zamal et al., 2012; Perozzi and Skiena, 2015; Aletras and Chamberlain, 2018) predicting user stance (Tan et al., 2011; West et al., 2014; Gryc and Moilanen, 2014; Gui et al., 2017; Wang et al., 2018; Volkova et al., 2014), and extracting inter-personal relations (Elangovan and Eisenstein, 2015; West et al., 2014; Abu-Jbara et al., 2013; Hassan et al., 2012). Most of these studies exploiting social network information are guided by an assumption of *homophily*, i.e., the tendency of individuals to associate and bond with similar others (McPherson et al., 2001). Our work differs from these studies in that we focus on classifying a given document (i.e. email) exchanged between users, not on predicting user information, nor interpersonal relations.

Note that different emails exchanged between the same set of users can belong to different classes, where in these studies, the attributes remains the same for a given set of users.

Graphs are an important data representation which occur naturally in various real-world applications, and graph analytics has been used in various tasks, including: node classification (Wang et al., 2017; Sen et al., 2008; Jian et al., 2018), link prediction (Wei et al., 2017; Pachev and Webb, 2017), and community detection (Fortunato, 2010; Cavallari et al., 2017).

Node embedding (a.k.a. graph or network embedding) aims to learn low-dimensional representations for nodes in graphs. Recently, network embedding methods have gained attention from the research community. Many recent node embedding models are inspired by neural language embedding models (Mikolov et al., 2013). These models include: *DeepWalk* (Perozzi et al., 2014), and *node2vec* (Grover and Leskovec, 2016). In these graph embedding models, a graph is represented as a set of sampled random walk paths. The embeddings for nodes then are learned in an unsupervised approach by applying the *word2vec* model (Mikolov et al., 2013) on the sampled paths. Hamilton et al. (2017b) categorize these models under *shallow learning* as they are inherently *transductive* and do not naturally generalize to unseen nodes. In our work, we are interested in applying models for email classification to new datasets.

GraphSAGE (Hamilton et al., 2017a; Hamilton, 2018) is an inductive graph embedding model. Unlike transductive models, it generalizes to unseen nodes and new graphs without requiring re-training. To do so, it learns a function that maps a node to low-dimensional representation by aggregating neighboring nodes' attribute information. We use GraphSAGE as a strong baseline for our email classification task. We discuss our usage of GraphSAGE in section 5..

## 2.2. Email Classification

Since the Enron corpus was made public, many researchers have used it for different tasks. Jabbari et al. (2006) released "the Sheffield dataset", in which they categorize a subset containing more than 12,000 Enron emails into two main categories "Business" and "Personal". Unlike our work, they do not utilize email thread structure, and many emails in the Sheffield dataset are not part of a thread and some threads are partially labeled (i.e. some emails in the thread are unlabeled). They also present a preliminary experiment for automatic classification of personal and business. We don't use this dataset for training as we are interested in modeling threads. However, we show the performance of some of our models on this dataset in subsection 7.2.. The Sheffield dataset has been used in other studies. In particular, Peterson et al. (2011) show that the formality level in emails is affected by the interpersonal nature of email (personal or business). They use email gold labels in the Sheffield dataset to determine the email type. Mitra and Gilbert (2012) use the Sheffield dataset to study the proportion of gossip in business and personal emails. In our work, we focus on automatic classification of emails into business and personal.

There has been some previous work on incorporating email communication network information for different tasks. Yoo et al. (2009) propose a semi-supervised method for personalized email prioritization. They find that including social features along with message content based features leads to a significant reduction in the prediction error when learning to identify the emails that a given user will consider important. Another task is to predict the recipient of an email. Graus et al. (2014) propose a generative model to predict the recipient of an email. They report that the optimal performance is achieved by combining features from both the communication graph and email content. Similar to our work, they use both Enron and Avocado. Our work is similar to (Wang et al., 2012) who propose a model for email classification into "Business" and "Personal". However, unlike our work, they don't use the email content. Their approach requires that the users (i.e. sender and recipients) have been seen in the labeled training data. Therefore, their approach cannot generalize to unseen users, let alone a new corpus (i.e. another email exchange). In contrast, our models do not require users to be seen before and can generalize to unseen nodes and new networks.

| Set | Business | Personal |
|---|---|---|
| Enron | 9,127 (86.7%) | 1,401 (13.3%) |
| Avocado | 4,810 (91.1%) | 467 (8.9%) |

Table 1: Distribution of Classes in the Datasets.

## 3. Corpus

We use the two datasets from our previous work (Alkhereyf and Rambow, 2017) that maintain the thread structure of emails (i.e. all emails belong to threads and all threads have labeled emails). The emails are taken from the well-known Enron email corpus, and the more recent Avocado corpus (Oard et al., 2015).

We split Enron into train, development and test sets with 50%, 25% and 25% of the emails respectively. We do not split threads. Avocado is divided equally into development and test sets (since we will not train on Avocado). Threads are chronologically ordered according to the time of the first email such that the training set contains the earliest threads and the test set contains the latest threads. We use subscripts $tr$, $dev$, and $ts$ to refer to the train, development and test sets respectively. We use the Enron$_{dev}$ for optimization.

Our Enron dataset contains 10,528 emails and Avocado contains 5,277 emails. Table 1 shows the distribution of "Business" and "Personal" emails in the datasets. In our experiments we optimize the personal F-1 score because our goal is to find the personal emails (minority class). Note that most of the threads in the corpus contain either only personal emails or only business ones. For Enron, there are 3,941 threads: 3,381 (85.8%) having business emails only, 450 (11.4%) having personal emails only, and 110 (2.8%) having mixed emails (i.e. some emails are business and others are personal). For Avocado, there are 1,975 total threads: 1,768 (89.5%) business only, 190 (9.6%) personal only, and 17 (0.9%) mixed.

## 4. Lexical Features

We use *FastText* (Bojanowski et al., 2017) to obtain word embeddings from the emails, which we use as lexical features. We use task-specific embeddings trained on the whole Enron email collection (not just our labeled subset). Both the body and subject are included in the training data. We use the CBOW mode with the default argument values. Arguments include the size of word vectors (100), the size of the context window (5), and the maximum and minimum length of n-grams (3 and 6, respectively).

To represent an email, we average the corresponding vectors for all character n-grams of every word in the email, then we compute the average vector for all words in the email (both the body and subject).

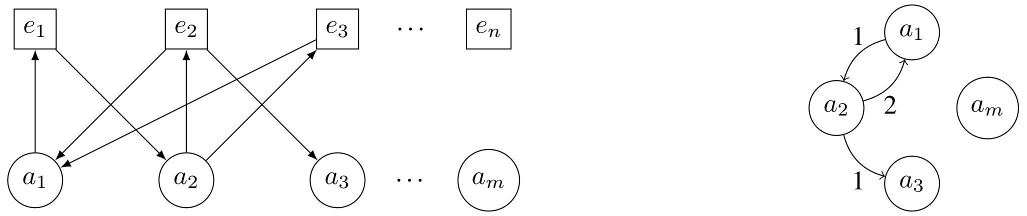
We have also tried various pre-trained *GloVe* (Pennington et al., 2014) vector sets that are available online, each trained using different corpora and embedded into various dimension sizes. We found that embeddings obtained using *FastText* from our data performed better than all pre-trained *GloVe* vector sets on all scores.

## 5. Baseline: Email Classification using GraphSAGE

GraphSAGE (Hamilton et al., 2017a) is a recent state-of-the-art inductive model for learning node embeddings for different tasks including node classification. It learns an embedding for a given node by aggregating information from its neighboring nodes and from attributes of the node. It is designed for homogeneous graphs where nodes belong to one type. Thus, we construct a graph which has only emails as nodes (We do not construct a graph with people as nodes since we also need access to the lexical content for GraphSAGE.) In this graph, nodes represent emails and edges link emails if they share a certain percentage of participants. We do not distinguish between senders and recipients as participants. Then, we feed the GraphSAGE supervised model with this graph of emails with their corresponding labels, and furthermore, we use the lexical features described in section 4. as node attributes.

We use the Jaccard similarity to measure the similarity between the participant sets of two emails and then link two emails with an edge if their similarity score is above a certain threshold. We define Jaccard similarity $J$ between two emails as:

$$J(e_i, e_j) = \frac{\mid \tau(e_i) \cap \tau(e_j) \mid}{\mid \tau(e_i) \cup \tau(e_j) \mid}$$



(a) Bipartite email-user graph

(b) User graph

Figure 1: Email exchange graphs.

Where $\tau(e_i)$ denotes the set of participants in email $e_i$ (both the sender and recipients). We experiment on Enron with different threshold values for $J(e_i, e_j)$ and report the one that optimizes the performance on the development set. Note that GraphSAGE implicitly models the thread structure, as emails in the same threads share the same participants, and thus are linked together in the graph.

## 6. Our Approach to Exploiting the Social Network

In this section, we present our approach to using the email network structure in our classification task. We start out by presenting two different ways of representing the social network induced by emails (subsection 6.1.). We then show how we derive features from these two types of graphs (subsection 6.2.). In subsection 6.3., we discuss an extension to GraphSAGE based on the bipartite graph we propose in subsection 6.1. and the features we extract in subsection 6.2.. Finally, in subsection 6.4., we propose a model that incorporates information from the thread structure of email into the prediction.

### 6.1. Graph Structures to Represent the Social Network

A very natural representation of the social network induced by email exchange is a bipartite graph with two disjoint sets of nodes: documents (i.e. emails) and users (i.e. people), such that there is an edge between an email and a user if and only if the user's email address appears as either the sender or a recipient (either in the "to" or "cc" list) in that email; we refer to this structure as the *bipartite email-user network*. Another option is a graph (not bipartite) whose nodes represent people (i.e. email addresses) and whose edges represent email communication such that an edge exists if there is at least one email that has been exchanged between the two end nodes; we refer to this structure as the *user network*. This graph is simply a one-mode projection of the bipartite graph. Figure 1 illustrates these two types of graphs. In both graphs we normalize multiple email addresses belonging to the same person into one user node. For each corpus (i.e. Enron and Avocado), we construct directed and undirected graphs from these two networks (i.e. the *bipartite email-user network* and the *user networks*). We use the whole exchange network, including all labeled and unlabeled emails to build these graphs.

In the directed bipartite network, each edge shows explicitly the directionality of the email (i.e. sender and recipients), while in the undirected bipartite graph, the directionality of communication is not reflected. The weights are always 1 in the bipartite graph. For the directed user network, edge directions indicate that the source user has sent emails to the target user, and the edge weight reflects the number of emails that have been sent from the source to the target, while in the undirected email network, edges indicate that the two connected nodes (i.e., users) have exchanged emails regardless of who sent the email.

### 6.2. Features Extracted from the Social Network

We extract different features from nodes in the corresponding directed and undirected graphs of both the bipartite email-user graph and the user graph. Some features are defined for only certain types of graphs (i.e. user vs. bipartite email-user; directed vs. undirected graphs), while other features are defined for all types of graphs. Then, we use these features with standard machine learning classifiers.

Table 2 shows all the social network features we use in our experiments. We have chosen the feature names to be as self-explanatory as possible. We divide them into three sets, as indicated by double horizontal lines in Table 2. First, node features that can be computed from its edges only. Second, features extracted by considering the node and its neighbors (i.e. adjacent nodes). Finally, for the third set, the values on a node feature depend on the node position in the whole graph. These three sets of features allow us to extract local and global properties of individual nodes.

**First feature set:** The *in-degree* and *out-degree* scores for a node indicate how many edges are directed to/from this node. For directed graphs, the *total degree* is the sum of these two numbers, and number of edges connected to the node in undirected graphs. For users, we extract this score from both the user graph and the bipartite graph. In the user graph, *in-degree* for a user is the number of other users who sent at least one email to this user, *out-degree* is the number of other users who received at least one email from this user, and the *total degree* indicates the number of people who have exchanged emails (sent or received) with this user. In the bipartite graph, *in-degree* score for a user node indicates how many emails have been received by this user and the *out-degree* indicates how many emails have been sent by this user. The total degree is the amount of all emails in which the user is participant in. For emails, *in-degree* is always equal to 1 (as any email always has only a single sender) so we ignore it. While out-degree indicates the number of recipients.

**Second feature set:** The second set of features measure dyadic relations and we extract them from the correspond-

| Feature | Directed Graph? | Undirected Graph? | User Graph? | Bipartite Graph? |
|---|---|---|---|---|
| In-, Out-Degree | ✓ | | ✓ | ✓ |
| Total Degree | ✓ | ✓ | ✓ | ✓ |
| # Common Neighbors | | ✓ | ✓ | |
| # Sender's triangles | | ✓ | ✓ | |
| # Common Neighbors/# Sender's triangles | | ✓ | ✓ | |
| Jaccard's coefficient | | ✓ | ✓ | |
| Clustering coefficient | | ✓ | ✓ | |
| In-, Out-degree centrality | ✓ | | ✓ | |
| Degree centrality | ✓ | ✓ | ✓ | |
| Betweenness centrality | ✓ | ✓ | ✓ | ✓ |
| Eigenvector centrality | ✓ | ✓ | ✓ | ✓ |
| Closeness centrality | ✓ | ✓ | ✓ | ✓ |
| Hub/Auth Score | ✓ | ✓ | ✓ | ✓ |

Table 2: Social Network Features. Check marks indicate that a feature is extracted only from the corresponding graph(s).

ing sender and recipient nodes of a given email in the user graph only. We extract these features for each pair of sender-recipient in case that an email has multiple recipients.

*Number of common neighbors* counts the common nodes shared between the sender and recipient(s). The number of common neighbors alone might not be a good indicator of how close a pair of users are in case that one of them is part of too many triangles. To overcome this issue, we calculate the *number of triangles* involving the sender. Then we use it as normalization factor for the number of common neighbors between the sender and recipient(s). The intuition is that if the sender has only a few triangles, then a high number of common neighbors indicates that the two users are well connected through common people. In contrast, a high number of triangles for the sender indicates that the sender is directly linked to many people who are linked to each other. We also compute *Jaccard's coefficient* score between the sender and recipient(s) which is simply the normalized number of common neighbors by the total neighbors (the union). The last feature in this set is the *local clustering coefficient*, which measures how close are neighbors for a given node to form a clique. We calculate *local clustering coefficient* for the sender and each recipient.

**Third feature set:** The last set of features measure the global importance of nodes in graphs. The *degree centralities* are the normalized degree scores (in, out and total) by the maximum possible degree. *Degree centralities* measure importance of a node by looking at its direct neighbors. This might be useful for users but not emails as there are important emails sent to a small number of users and less important emails sent to many users (e.g. announcements). Thus, we compute them only for users in the user graph. Other centrality measures (*betweenness*, *eigenvector*, and *closeness centralities, hub/auth*) take into account nodes other than direct neighbors. Each centrality score computes the importance of a node differently. Particularly, *closeness centrality* indicates how close a node is to all other nodes in the network. It is the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. While *betweenness centrality* measures the number of times a node lies as a bridge on the shortest path between two other nodes. All of these scores do not take into account the importance of the other nodes. For instance, a node might be connected (or acts as a bridge) to a few but important nodes but has a lower score than another node which is connected to a lot of less important nodes. To overcome this issue, we use *eigenvector centrality*. It measures the importance of a node by taking into account the importance of other nodes. A high eigenvector score means that a node is connected to many nodes who themselves have high scores. *Hub/Auth* is a generalization of *eigenvector centrality*. For each node, we compute two scores: hub score and authority score. A high hub score for a nodes means that it points to nodes with high authority scores. While a high authority score means the node is being connected by nodes with high hub scores.

We compute these scores for both user (sender and recipients) and email nodes in both the bipartite and user graphs.

**Final network feature vector:** As we are interested in classifying emails, we extract features corresponding to emails and their participants. For each email, we extract features described above from the corresponding email node in the bipartite email-user graph as well as features from both the sender and the recipients (either in the "to" or "cc" list) from both the user graph and the bipartite email-user graph. In case the email has multiple recipients, we compute the max, min and average of the value corresponding to each feature. We then feed these features to machine learning models.

### 6.3. GraphSAGE with Bipartite Graph

Our baseline, GraphSAGE (section 5.), is not designed to deal with the heterogeneous network induced by email exchange that includes emails and participants. Therefore, we extend GraphSAGE as follows. We construct a bipartite graph of users and emails as discussed in subsection 6.1.. Then, we feed this graph to a version of GraphSAGE which we modified such that we have different encoders and aggregates for users and emails. For emails, we use lexical features to represent them. For users, we use the network features extracted from the corresponding node in the user graphs as discussed in subsection 6.2.. We refer to this method as *GraphSAGE-BiP*. Because of the extension to bipratite graphs and the use of our own features, *GraphSAGE-BiP* represents a contribution of this paper.

### 6.4. Sequential Modeling of Threads

In the previous subsections, we have presented models on individual emails without looking to other emails in the same thread. However, we can predict the class of an email from other emails in the same thread; in fact, we observe that only 2.8% of threads in our Enron data set contain both "Personal" and "Business" emails.

In this subsection, we discuss how we incorporate information from other emails in the same thread in order to improve the classification. We try two methods: first, using sequential models on threads, namely, LSTMs; and second, we add a simple approach that re-predicts email labels based on the majority of the predicted email labels in the same thread.

**Modeling threads using LSTMs** We apply Long Short Term Memories (LSTMs) networks to model thread structure. We concatenate two Bidirectional LSTMs (BiLSTMs), one for lexical features and the other for the social network features. Figure 2 illustrates the model architecture.

**Majority of the thread** We first predict emails using LSTMs. Then, we compute the majority vote of all emails in the same thread and assigning the majority label to each email in the thread. In case that there is no majority (i.e. the numbers of predicted business and personal labels are the same), we consider "Personal" to be the majority label.

## 7. Experiments

In this section, we present experimental results of the email classification task into "Business" and "Personal" by conducting different experiments in different settings. In these experiments, we optimize the F-1 score on Personal emails since we are trying to identify personal emails, which are rare. We also report accuracy and Business F-1, along with recall and precision, since all measures together give a more complete understanding of the performance of our classifiers. In our results, we report the model with the optimal hyper-parameters that maximize the Personal F-1 score.

In the following subsections, we first define weak baselines in subsection 7.1.. Then we evaluate some models on the Sheffield data set (Jabbari et al., 2006) in subsection 7.2.. In subsection 7.3., we evaluate different models and feature sets on individual emails without looking to other emails in the same thread. In subsection 7.4., we discuss the results of models for sequential modeling of threads. Finally, we discuss performance on the test set (subsection 7.5.). Table 5 summarizes the results.

### 7.1. Weak Baselines

An addition to our strong baseline, GraphSAGE (section 5.), we define two weak baselines: a random classifier, and the all-business classifier. The former predicts the classes by respecting the class distribution in the Enron training dataset, while the latter predicts the majority class (i.e. "business"). Table 4 shows the results of these two baselines on our datasets. While the random baseline can be compared against the performance of our models on the minority class ("Personal"), for the all-business baselines, the personal F-1 score could be trivially beaten (zero score). However, it is harder to beat the business F-1 score of the all-business baseline, since the datasets are highly unbalanced (all datasets have more than 80% business emails). We consider a model robust if it has a personal F-1 score higher than random and a business F-1 score higher than all-business.

| | | Bus | | | Pers | | |
|---|---|---|---|---|---|---|---|
| Model | Acc | F1 | Rec | Prec | F1 | Rec | Prec |
| shf | 93 | 95 | 99 | 92 | 80 | 69 | 95 |
| net | 86.2 | 90.2 | 87.4 | 93.1 | 77.2 | 83.2 | 72.0 |
| lex | 95.3 | 96.7 | 96.8 | 96.7 | 91.6 | 91.4 | 91.8 |
| all | 96.0 | 97.2 | 97.6 | 96.9 | 92.7 | 91.8 | 93.6 |

Table 3: Results of our models on the Sheffield dataset. We show numbers reported in (Jabbari et al., 2006) as (shf); their results are not directly comparable and are only shown for rough benchmarking.

### 7.2. Evaluation on Sheffield Data

In this subsection, we evaluate SVM classifiers on the Sheffield dataset (subsection 2.2.). The information about the experiments described in Jabbari et al. (2006) is not detailed and does not mention the train and test ratios. We divide the Sheffield set into 75% and 25% for train and test respectively. Table 3 shows results of three SVM classifiers: with network features only, with lexical features only, and with combination of both features (see subsection 7.3. for details). In addition, we report the results of the preliminary experiment reported in Jabbari et al. (2006) for convenience. However, the results are not directly comparable as we do not know what their training data was. The results show that our models outperform the results in Jabbari et al. (2006). Moreover, it shows that incorporating social network features with the lexical features outperforms modeling emails with lexical features only.

### 7.3. Classifying Emails Individually

In this subsection, we evaluate different models using individual emails without looking to other emails in the same thread.

We experiment with three classifiers: Deep Neural Networks (DNNs), Support Vector Machines (SVMs) and GraphSAGE-BiP (see subsection 6.3.). For DNNs, we use feed-forward neural networks and we try different hyperparameters (i.e. number of hidden units, and number of layers). We try linear and RBF kernels for SVMs. We tune the hyperparameters on Enron$_{dev}$.

For the SVM and NN classifiers we use three feature sets: **net**, using social network features only (section 6.); **lexical**, using word embeddings only (section 4.); **all**, the combination of the two feature sets. In the **all** feature setting, for neural networks, we concatenate the two networks (branches) of the lexical and the network features. For SVMs, we take the average of the two kernels (a kernel for each feature set).

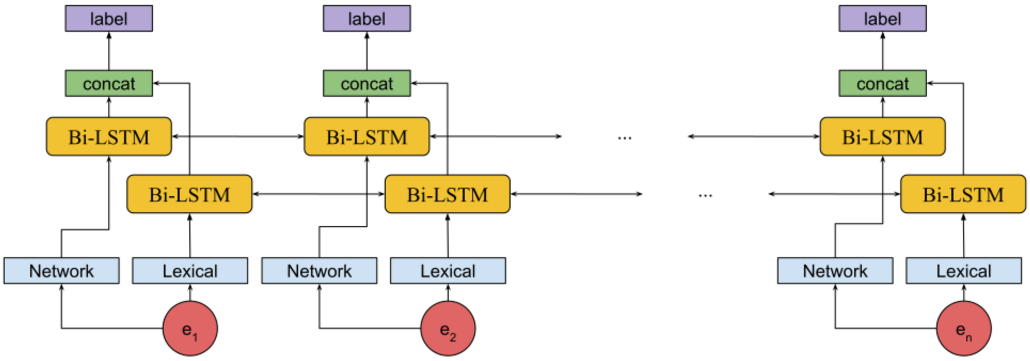


Figure 2: Two concatenated BiLSTMs for thread sequential modeling; one for lexical features and the other for social network features.

| | | DEV | | | TEST | | |
|---|---|---|---|---|---|---|---|
| Baseline | Set | Accuracy | Business F-1 | Personal F-1 | Accuracy | Business F-1 | Personal F-1 |
| Expected Random | Enron | 77.1 | 86.8 | 13.2 | 76.7 | 86.5 | 13.5 |
| | Avocado | 80.5 | 89.0 | 10.4 | 80.2 | 88.8 | 10.7 |
| All-Business | Enron | 86.8 | 92.9 | 0 | 86.2 | 92.6 | 0 |
| | Avocado | 91.4 | 95.5 | 0 | 90.9 | 95.3 | 0 |

Table 4: Results of different baselines trained on Enron$_{tr}$ and tested on the indicated set. Here, we report the expected values for the random classifier.

| | **DEV** | | | | | | | **TEST** | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Acc | Bus | | | Pers | | | Acc | Bus | | | Pers | | |
| Model | | F1 | Rec | Prec | F1 | Rec | Prec | | F1 | Rec | Prec | F1 | Rec | Prec |
| **Enron** | | | | | | | | | | | | | |
| GS | 91.10 | 94.81 | 93.56 | 96.09 | 68.97 | 74.92 | 63.89 | 88.77 | 93.30 | 90.66 | 96.08 | 65.43 | 76.95 | 56.91 |
| svm-net | 85.39 | 91.67 | 92.62 | 90.74 | 40.56 | 37.79 | 43.77 | 83.99 | 90.81 | 91.81 | 89.84 | 37.79 | 35.2 | 40.79 |
| svm-lex | 91.02 | 94.76 | 93.61 | 95.94 | 68.48 | 73.94 | 63.76 | 89.5 | 93.82 | 92.41 | 95.27 | 65.24 | 71.34 | 60.1 |
| svm-all | 91.45 | 95.02 | 93.96 | 96.1 | 69.8 | 74.92 | 65.34 | 89.59 | 93.86 | 92.41 | 95.36 | 65.63 | 71.96 | 60.31 |
| nn-net | 84.66 | 91.22 | 91.83 | 90.62 | 39.18 | 37.46 | 41.07 | 83.99 | 90.8 | 91.61 | 90.0 | 38.61 | 36.45 | 41.05 |
| nn-lex | 91.53 | 95.09 | **94.46** | 95.74 | 69.27 | 72.31 | 66.47 | 89.29 | 93.74 | 93.06 | 94.43 | 62.89 | 65.73 | 60.29 |
| nn-all | 91.79 | 95.23 | 94.36 | 96.12 | 70.66 | 74.92 | **66.86** | 89.97 | 94.13 | 93.21 | 95.06 | 65.79 | 69.78 | 62.22 |
| GS-BiP | 91.41 | 94.99 | 93.86 | 96.15 | 69.79 | 75.24 | 65.07 | 89.89 | 94.07 | 93.11 | 95.06 | 65.59 | 69.78 | 61.88 |
| LSTM | **91.88** | **95.26** | 94.11 | 96.45 | 71.49 | 77.2 | 66.57 | 90.75 | 94.56 | **93.36** | 95.80 | 69.98 | 75.45 | **64.25** |
| LSTM+ | 91.71 | 95.14 | 93.61 | **96.73** | **71.58** | **79.15** | 65.32 | **91.03** | **94.71** | 92.87 | **96.64** | **70.49** | **79.3** | 63.44 |
| **Avocado** | | | | | | | | | | | | | |
| GS | 90.15 | 94.41 | 91.13 | **97.95** | 58.33 | **79.82** | 45.96 | 88.30 | 93.36 | 89.66 | 97.37 | 54.19 | **75.73** | 42.19 |
| svm-net | 86.4 | 92.41 | 90.63 | 94.26 | 34.61 | 41.67 | 29.6 | 84.79 | 91.42 | 89.07 | 93.89 | 33.28 | 41.84 | 27.62 |
| svm-lex | 91.44 | 95.21 | 93.16 | 97.36 | 59.64 | 73.25 | 50.3 | 89.84 | 94.28 | 92.16 | 96.51 | 54.27 | 66.53 | 45.82 |
| svm-all | 92.27 | 95.7 | 94.3 | 97.1 | 61.2 | 70.6 | 54.0 | 90.75 | 94.8 | 93.3 | 96.4 | 56.1 | 65.3 | 49.2 |
| nn-net | 88.48 | 93.75 | 94.57 | 92.95 | 26.57 | 24.12 | 29.57 | 88.28 | 93.64 | **94.79** | 92.51 | 26.25 | 23.01 | 30.56 |
| nn-lex | 92.54 | 95.88 | 95.07 | 96.71 | 60.36 | 65.79 | 55.76 | 90.9 | 94.96 | 94.25 | 95.68 | 53.31 | 57.32 | 49.82 |
| nn-all | 92.99 | 96.13 | 95.23 | 97.04 | 63.07 | 69.3 | 57.88 | 91.32 | 95.19 | 94.54 | 95.86 | 55.19 | 59.0 | 51.84 |
| GS-BiP | 91.10 | 95.09 | 93.12 | 97.02 | 57.50 | 69.74 | 48.92 | 90.97 | 94.94 | 93.08 | 96.88 | 58.39 | 69.87 | 50.15 |
| LSTM | **93.67** | **96.50** | **95.48** | 97.54 | 67.06 | 74.56 | **60.93** | **91.77** | **95.40** | 93.87 | 96.98 | 60.90 | 70.71 | **53.48** |
| LSTM+ | 93.64 | 96.47 | 95.07 | 97.91 | **68.06** | 78.51 | 60.07 | 91.66 | 95.31 | 93.29 | **97.43** | **62.07** | 75.31 | 52.79 |

Table 5: Results for all models on Enron and Avocado using different classifiers with different feature sets. All models are trained only on Enron$_{tr}$. GS is the GraphSage baseline. The SVM, NN (neural network), and GS-BiP models (GraphSage with our extension to bipartite graphs) model emails individually (without thread structure). For SVM and NN results, we give results with different feature sets, namely net (social network features only), lex (lexical feature only), and all (all features). The LSTMs model the thread structure explicitly. LSTM+: LSTM with majority vote

Table 5 shows the results of models for email classification on both corpora: Enron and Avocado. In the first line, we report the results for our baseline, GraphSAGE (GS). We then present the results for our experiments using SVM and NN, each with the three possible feature sets. Finally, we present the results for our version of GraphSAGE using bipartite graphs, GS-BiP.

To determine whether the performance improvement of different classifiers over others is statistically significant, we use the non-parametric Wilcoxon Signed-Rank Test (Sidney, 1957) on pairs of the Personal F-1 scores of different classifiers using 10 fold-cross validation runs on Enron. We perform the test on some crucial results, and we report the results of *all* significance tests we have performed, whether successful or not.

We observe that all models beat the random baseline on both Bus F1 and Pers F1 scores. However, classifiers with the network features alone perform worse than the all-business classifier on the business F1 score on both corpora. Other classifiers (i.e. lex and all) outperform the all-business classifier on Enron while only a few individual email modeling classifiers have higher Bus F1 scores than the all-business classifier on Avocado.

In general, lexical features alone outperform the network features alone. However, for all models on both corpora, incorporating social network information with lexical features improves the performance over the lexical features alone. For SVMs on Enron, this increase is significant at $p < 0.01$. Note that the neural model also profits from the addition of "feature engineered" network features.

For GraphSAGE on Enron, we performed an additional experiment (we do not give full results) in which we remove the network information by simply creating a graph without any edges between the nodes that represent emails. This amounts to just using lexical information in creating the node embeddings. Using lexical information only in this manner does not significantly decrease the results over using the network structure in conjunction with lexical information. We conclude that GraphSAGE does not succeed in exploiting the information in the network induced by emails, while our feature-based approach to the network structure does.

The SVM-all and NN-all models both beat GraphSAGE and GraphSAGE-BiP on Enron by a small margin (the difference is statistically significant for Personal F-1). Furthermore, as expected, the NN models outperform the SVM models (recall that both models use exactly the same features). We observe that the extension of GraphSAGE to bipartite graph (GS-BiP) outperforms GraphSAGE using homogeneous graphs.

The results also show that the performance in the inter-corpora setting is lower than the performance in the intra-corpus setting for both social network and lexical features, for all models. We observe that GraphSAGE performs much worse in the inter-corpora setting compared to the intra-corpus. In addition, in the intra-corpus setting, the network features add more improvement. This is expected since Enron and Avocado have different email graphs and different professional languages (Enron was an energy company and Avocado was an IT company operating a decade later).

These observations and results suggest that incorporating social network information with the lexical features indeed improves the performance in our approach. Also, the models can generalize to a new corpus without the need for retraining, and the network features play an important role in the performance on a new corpus.

### 7.4. Classifying Emails in Threads

The last two lines for each corpus in Table 5 show the results of LSTMs only and LSTMs with majority vote (LSTM+). The results show that LSTMs models perform better than models trained on individual emails on both the personal and business F-1 scores. Also, they beat the All-business baseline on the Bus F1 score, which makes them robust classifiers. The improvements of LSTMs over best non-sequential models on both corpora (NN-all and GraphSAGE-BiP) are statistically significant ($p < 0.01$). We observe that applying majority vote to LSTM models increases the personal F-1 score but in some cases decreases the business F-1. We also observe that using LSTMs increases the performance across the board, but the increase is particularly marked for the testing on Avocado. This reflects that the LSTM can exploit similarities among emails of a thread. We also note that the LSTM model with majority vote outperforms the GraphSAGE model by a substantial margin, providing our best results.

### 7.5. Performance on the Test Set

The results on the blind test set mirror, by and large, the results on the dev set.

We observe a drop in the performance for both test sets in comparison to the corresponding development set. For Enron, we expect a slight decrease in the results since we optimize our models on the development set. However, for Avocado, we have not optimized any of our models on the Avocado development set. This suggests that Avocado$_{ts}$ is just harder than Avocado$_{dev}$. Note that the sizes of Avocado$_{dev}$ and Avocado$_{ts}$ are almost the same and their ratio of personal emails is very similar: $8.6\%$ and $9.1$, respectively.

## 8. Conclusion

In this paper, we propose a new way of incorporating social network information from the underlying email exchange network for email classification into "Business" and "Personal". In addition, we use a state-of-the-art graph embedding model namely, GraphSAGE, as a strong baseline. Our main finding is that adding social network information to lexical features improves the classification performance over the performance of an approach based on textual information only. Our models beat the strong baseline. We also find that modeling the thread structure improves the classification performance further, giving a substantial boost over GraphSAGE. The results also show that our network features can generalize to unseen nodes and graphs as we train on the email of one company (Enron) and test on the emails of another company (Avocado) that has different email graphs. We suggest that generic graph embedding models such as GraphSAGE are powerful tools for exploiting the social network, but they don't always have the best performance on some tasks. More importantly, the results of the extension of GraphSAGE to bipartite graph (i.e. GS-BiP) suggest that the choice of graph representation of the communication network is crucial for the classification performance, and requires changes to the GraphSAGE algorithm.

For future work, we intend to experiment with combining our approach with the GraphSAGE embeddings. Our methodology of incorporating social network information is not limited to email classification, and we intend to investigate other interpersonal document classification tasks.

## Acknowledgments

We would like to thank the anonymous reviewers for their constructive feedback. The first author is sponsored by the KACST graduate scholarship program.

## 9. Bibliographical References

Abu-Jbara, A., King, B., Diab, M., and Radev, D. (2013). Identifying opinion subgroups in Arabic online discussions. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 829–835.

Agarwal, A., Omuya, A., Harnly, A., and Rambow, O. (2012). A comprehensive gold standard for the Enron organizational hierarchy. In *50th Annual Meeting of the Association for Computational Linguistics*, page 161.

Al Zamal, F., Liu, W., and Ruths, D. (2012). Homophily and latent attribute inference: Inferring latent attributes of Twitter users from neighbors. *ICWSM*, 270:2012.

Aletras, N. and Chamberlain, B. P. (2018). Predicting Twitter user socioeconomic attributes with network and language information. In *Proceedings of the 29th on Hypertext and Social Media*, pages 20–24. ACM.

Alkhereyf, S. and Rambow, O. (2017). Work hard, play hard: Email classification on the Avocado and Enron corpora. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 57–65.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Cavallari, S., Zheng, V. W., Cai, H., Chang, K. C.-C., and Cambria, E. (2017). Learning community embedding with community detection and node embedding on graphs. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 377–386. ACM.

Elangovan, V. K. and Eisenstein, J. (2015). "You're Mr. ebowski, I'm the dude": Inducing address term formality in signed social networks. In *HLT-NAACL*, pages 1616–1626.

Filippova, K. (2012). User demographics and language in an implicit social network. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1478–1488. Association for Computational Linguistics.

Fortunato, S. (2010). Community detection in graphs. *Physics reports*, 486(3-5):75–174.

Graus, D., Van Dijk, D., Tsagkias, M., Weerkamp, W., and De Rijke, M. (2014). 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. ACM.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.

Gryc, W. and Moilanen, K. (2014). Leveraging textual sentiment analysis with social network modelling. *From Text to Political Positions: Text analysis across disciplines*, 55:47.

Gui, L., Zhou, Y., Xu, R., He, Y., and Lu, Q. (2017). Learning representations from heterogeneous network for sentiment classification of product reviews. *Knowledge-Based Systems*, 124:34–45.

Hamilton, W., Ying, Z., and Leskovec, J. (2017a). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1025–1035.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017b). Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Hamilton, W. L. (2018). *Representation Learning Methods for Computational Social Science*. Ph.D. thesis, Stanford University.

Hassan, A., Abu-Jbara, A., and Radev, D. (2012). Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 59–70. Association for Computational Linguistics.

Jabbari, S., Allison, B., Guthrie, D., and Guthrie, L. (2006). Towards the Orwellian nightmare: separation of business and personal emails. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 407–411. Association for Computational Linguistics.

Jian, L., Li, J., and Liu, H. (2018). Toward online node classification on streaming networks. *Data Mining and Knowledge Discovery*, 32(1):231–257.

Kiritchenko, S. and Matwin, S. (2011). Email classification with co-training. In *Proceedings of the 2011 Conference of the Center for Advanced Studies on Collaborative Research*, pages 301–312. IBM Corp.

McPherson, M., Smith-Lovin, L., and Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mitra, T. and Gilbert, E. (2012). Have you heard?: How gossip flows through workplace email. In *ICWSM*.

Pachev, B. and Webb, B. (2017). Fast link prediction for large networks using spectral embedding. *Journal of Complex Networks*, 6(1):79–94.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Perozzi, B. and Skiena, S. (2015). Exact age prediction in social networks. In *Proceedings of the 24th International Conference on World Wide Web*, pages 91–92. ACM.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.

Peterson, K., Hohensee, M., and Xia, F. (2011). Email formality in the workplace: A case study on the Enron corpus. In *Proceedings of the Workshop on Languages in Social Media*, pages 86–95. Association for Computational Linguistics.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., and Eliassi-Rad, T. (2008). Collective classification in network data. *AI magazine*, 29(3):93.

Sidney, S. (1957). Nonparametric statistics for the behavioral sciences. *The Journal of Nervous and Mental Disease*, 125(3):497.

Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., and Li, P. (2011). User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM.

Volkova, S., Coppersmith, G., and Van Durme, B. (2014). Inferring user political preferences from streaming communications. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 186–196.

Wang, M.-F., Tsai, M.-F., Jheng, S.-L., and Tang, C.-H. (2012). Social feature-based enterprise email classification without examining email contents. *Journal of Network and Computer Applications*, 35(2):770–777.

Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., and Yang, S. (2017). Community preserving network embedding. In *AAAI*, pages 203–209.

Wang, H., Zhang, F., Hou, M., Xie, X., Guo, M., and Liu, Q. (2018). Shine: signed heterogeneous information network embedding for sentiment link prediction. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 592–600. ACM.

Wei, X., Xu, L., Cao, B., and Yu, P. S. (2017). Cross view link prediction by learning noise-resilient representation consensus. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1611–1619. International World Wide Web Conferences Steering Committee.

West, R., Paskov, H. S., Leskovec, J., and Potts, C. (2014). Exploiting social network structure for person-to-person sentiment analysis. *Transactions of the Association for Computational Linguistics*, 2:297–310.

Yoo, S., Yang, Y., Lin, F., and Moon, I.-C. (2009). 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. ACM.

## 10. Language Resource References

Apoorv Agarwal and Sakhar Alkhereyf and Adinoyi Omuya and Aaron Harnly and Owen Rambow. (2020). *Gender, Power, Business/Personal Type Annotations for the Enron Email Corpus*. Columbia University, ISLRN 903-314-357-253-8.

Sakhar Alkhereyf and Owen Rambow. (2020). *Business/Personal Type Annotations for the Avocado Email Corpus*. Columbia University, ISLRN 528-821-149-515-9.

Douglas Oard and William Webber and David Kirsch and Sergey Golitsynskiy and Douglas Reynolds. (2015). *Avocado Research Email Collection*. Linguistics Data Consortium, ISLRN 102-408-869-995-0.