Work Hard, Play Hard: Email Classification on the Avocado and Enron Corpora

Sakhar Alkhereyf
Department of Computer Science
Columbia University
New York, U.S.A.
sakhar@cs.columbia.edu

Owen Rambow
Center for Computational Learning Systems
Columbia University
New York, U.S.A.
rambow@ccls.columbia.edu

Abstract

In this paper, we present an empirical study of email classification into two main categories "Business" and "Personal". We train on the Enron email corpus, and test on the Enron and Avocado email corpora. We show that information from the email exchange networks improves the performance of classification. We represent the email exchange networks as social networks with graph structures. For this classification task, we extract social networks features from the graphs in addition to lexical features from email content and we compare the performance of SVM and Extra-Trees classifiers using these features. Combining graph features with lexical features improves the performance on both classifiers. We also provide manually annotated sets of the Avocado and Enron email corpora as a supplementary contribution.

1 Introduction

Email has quickly become a crucial communication medium for both individuals and organizations. Kiritchenko and Matwin (2011) show that a typical user daily receives 40-50 emails. Because of its popularity, different research problems related to email classification tasks have arisen. These tasks include spam-filtering, assigning priority to messages, and foldering messages according a user-specified strategy (Klimt and Yang, 2004). In spite of the popularity of email, many classification tasks have been hampered due the lack of availability of task-related data, due to the privacy issues surrounding email. However, two large data sets are available. First, a large dataset of real emails, the Enron corpus, was made publicly available by the Federal Energy Regulatory Commission (FERC) during the legal investigation of the company's collapse. Second, in February 2015, the Linguistic Data Consortium distributed a data set of emails from an anonymous defunct information technology company referred as Avocado (Oard et al., 2015).

In this paper, we present an empirical study on email classification into two categories: Business and Personal. We train only on the Enron corpus, but test on both the Enron and Avocado corpora for this classification task in order to investigate how dependent on the training corpus the learned models are. In addition, we provide new annotated datasets based on the two corpora [1].

We manually annotated datasets based on the Enron and Avocado corpora for this classification task. We use lexical features as well as social network features extracted from the email exchange network of both Enron and Avocado. The experiments show that when the social network features combined with lexical features outperforms the lexical features alone.

We first present some related work on both the Enron and Avocado corpora (Section 2). Then in Section 3, we describe the datasets and the annotation scheme used in this paper. We discuss lexical features in Section 4, and show how to extract social network features from the email exchange in Section 5. Finally, we present some experiments with different settings (Section 6). The experiments show that adding features extracted from graphs of the email exchange to the lexical features improves the classification performance.

[1] http://www.cs.columbia.edu/~sakhar/resources.html

Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing, ACL 2017, pages 57–65,
Vancouver, Canada, August 3, 2017.

2 Related Work

Since the Enron corpus has been made publicly available, many researchers have worked on the Enron corpus with different tasks. To our knowledge, the previous effort most closely related to this paper is that of Jabbari et al. (2006). They released a large set of manually annotated emails, in which they categorize a subset of more than 12,000 Enron emails into two main categories: "Business" and "Personal" and then into subcategories "Core Business" and "Close Personal". These sub-categories represent the two main categories respectively. The "Core Business" category has more than 4,500 emails while the "Close Personal" has more than 1,800. We compare our data to their data in detail in Section 3.

Agarwal et al. (2012) released a gold standard of the Enron power hierarchy and predict the dominance relations between two employees using the degree centrality of the email exchange network. They released this gold standard of the Enron corpus with thread structure as a MongoDB database. Hardin et al. (2014) study the relation between six social network centrality measures and the hierarchal ranking of Enron employees.

Mitra and Gilbert (2013) study gossip in the Enron corpus. They use the data set in Jabbari et al. (2006) to study the proportion of gossip in business and personal emails and find that gossip appears in both personal and business emails and at all levels of the organizational hierarchy. They use an NER classifier to label person names in emails then classify emails mentioning a person not in the recipient list nor the sender as gossip.

A related task is to predict the recipient of an email. Graus et al. (2014) propose a generative model to predict the recipient of an email using the email communication graph and the email content. The model is trained on Enron and tested on Avocado. The full enterprise email exchange network is used to build the communication graph as a directed graph, as we do in Section 5. They report that the optimal performance is achieved by combining the communication graph and email content.

3 Datasets and Annotation Scheme

As a part of the work in this paper, we have used the Amazon Mechanical Turk (AMTurk) crowdsourcing platform to annotate a subset of the Enron corpus. In addition, due to license constraints, we have in-house annotated a subset of the Avocado corpus. We use these two sets as well as the dataset distributed by Jabbari et al. (2006) (which we refer to as the "Sheffield set") for the classification task in this paper.

3.1 Labeling

Unlike Jabbari et al. (2006), we are interested in maintaining the thread structure of emails (for future work). Annotators were given email threads of various lengths and asked to annotate each email in the thread and to annotate the thread as a whole. However, classifying email content into business and personal can be a subjective task. For example, if an email talks about an invitation to a picnic for the employees families, one annotator might label this email as business email with the perspective that it talks about a business-related event. On the other hand, another annotator might have a perspective that this is personal event even though it is organized by the company.

We have provided instructions for the annotators to annotate each email with one of the following labels and criteria:

1. Business: The content of the message is clearly professional (even if the language used is very friendly) and it does not contain any personal content; it should be related to the company work.

2. Somehow Business: The main purpose of the message is professional but it has some personal parts.

3. Mixed: the content of the message belongs to two or more of the categories (typically because the sender combines different content in one email).

4. Somehow Personal: The main purpose of the message is personal but it has some business-related content.

5. Personal: The content of the message is clearly personal (even if the language used is very formal) and it does not contain any professional part.

6. Cannot Determine: If the there is no enough content to determine the category.

We added some detailed instructions to deal with certain cases:

	Threads			Emails		
Set	Business	Personal	Total	Business	Personal	Total
Enron$_T$	3,101 (82.8%)	642 (17.2%)	3,743	9,145 (86.7%)	1,401 (13.3%)	10,546
Sheffield$_{all}$	NA	NA	NA	9,857 (75.7%)	3,168 (24.3%)	13,025
Sheffield$_{sub}$	NA	NA	NA	4,525 (73.7%)	1,611 (26.3%)	6,136
Enron$_{\cap A}$	NA	NA	NA	2,513 (88%)	342 (12%)	2,855 (88.6%)
Enron$_{\cap D}$	NA	NA	NA	NA	NA	367 (11.4%)
Enron$_{\cap}$	NA	NA	NA	NA	NA	3,222
Enron$_{\cup}$	NA	NA	NA	16,223 (79.7%)	4,126 (20.3%)	20,349

Table 1: Summary of the Enron datasets

- If a message is about a social event inside the company, such as celebrating a new baby of an employee, or a career promotion, it belongs to the second category ("somehow business").
- If a message is about a social event outside the company but still related to the company, such as a picnic (usually family members are invited), it belongs to the fourth category ("somehow personal").
- If a message is about a social event which is not related to the company such as a charity but company employees are encouraged to participate, it belongs to the fourth category ("somehow personal").
- If a message is too short to determine its category (or even empty), it should have the same category as the message it is responding to, or the message it is forwarding.
- If a message is ambiguous, try to read other messages in the thread to clarify.
- If a message is spam or in the rare case that the first message of a thread is very short or empty, say "cannot determine".

3.2 Annotators

In the AMTurk task (i.e. Enron), each email thread was annotated by three different turkers. The group of turkers differs from a thread to another. We first ran several batches on AMTurk in which we assigned 5 annotators to each HIT; by studying the resulting data sets, we found that 3 annotators is sufficient and less costly, and most of the data was annotated using 3 Turkers.

To determine the consensus label, we give each of the categories in the above list a numerical label between 1 and 6, with 6 being "cannot determine" and otherwise a larger number indicating that the email is more personal. First, we discard any "cannot determine" label. Therefore, if there is one or more labels other than "cannot determine" we limit voting to these labels. If all labels are "cannot determine", the result of voting is "cannot determine" too. Then, we compute the majority vote of all labels from the three turkers, in case of ties, we take the floor of the mean of ties. For instance, if the labels are $\{1, 2, 6\}$ the majority vote result is $\{1, 2\}$. The mean is 1.5 and the floor is 1. Thus, the final label is 1. There are 5,372 (50.8%) emails in which all annotators gave the same label. The number of emails for each category with consensus among all annotators as follows:

Business	4,882
Somehow Business	17
Mixed	8
Somehow Personal	438
Personal	0
Cannot Determine	27

The average standard deviation of ordinal values (i.e. 1: business, 2: Somehow Business ... etc) in Enron emails $= 0.37$. For computing the average of standard deviation, we exclude any "Cannot Determine" label before computing the standard deviation per email, and if the email has less than two labels other than "Cannot Determine", we exclude that email too. We do so because "Cannot Determine" has no actual ordinal value.

For the annotation of the Avocado corpus, we hired two in-house undergraduate students to annotate two overlapping subsets of the Avocado corpus, using the same instructions as we gave the Turkers. The licensing conditions for this corpus appear to prohibit using AMTurk. In case of disagreement in Avocado$_{\cup}$ (described in 3.4), we arbitrarily choose the first annotator's label for consistency, unless the first is "cannot determine", in which case we choose the second. The average standard deviation of ordinal values (i.e. 1: business, 2: Somehow Business ... etc) in Avocado emails $= 0.08$. Since we have only two annota-

tors, we exclude any email labeled "Cannot Determine" by any annotator. The inter-annotator agreement in Avocado emails $\kappa = 0.58$ (Cohen's kappa). [2]

The complex labeling scheme described here will be useful for different tasks in the future. However, for the goal of this paper, we aim to group these labels into binary classes: business and personal. Therefore, we normalize the labels as follows: we group "Business" and "Somehow Business" into one category "Business", and "Personal", "Somehow Personal" and "Mixed" into one category "Personal". "Cannot Determine" remains the same.

Finally we exclude emails with labels other than "Business" or "Personal" (i.e. emails labeled as "Cannot determine"). These emails are discarded in both training and evaluation. This label is very rare; it occurs only 0.26% of the time in the Enron data, and 0.38% in the Avocado data.

3.3 Enron Datasets

The annotated emails by turkers are a subset of the Enron corpus released by Agarwal et al. (2012), which has more than 36,000 threads and 270,000 emails. We choose this version of Enron because it maintains the thread structure of emails. From this collection, we have randomly sampled total of 3,941 threads with different numbers of emails per thread (2, 3, 4, and 5). The total number of emails is 10,573. We exclude 198 threads (5%) and 27 additional emails (0.26%) labeled as "Cannot determine". The sample has 3,222 emails overlapping with the Sheffield set of Jabbari et al. (2006) (after excluding "Cannot determine" emails). We also exclude all emails in the Sheffield set that we could not match with an email in (Agarwal et al., 2012). After obtaining the final labels as described in 3.2, we got 3,743 threads and 10,546 emails labeled as either "Business" or "Personal" from the Enron corpus. Table 1 shows the summary of the Enron datasets with the following notations:

- Enron$_T$: The threads and emails obtained from AMTurk as in 3.2.
- Sheffield$_{all}$: All the Sheffield set except those that we could not match in (Agarwal et al., 2012).
- Sheffield$_{sub}$: A subsample of the the Sheffield set ("Business Core" and "Personal Close").
- Enron$_{\cap A}$: The intersection between Enron$_T$ and Sheffield$_{all}$ in which both agree in labels.
- Enron$_{\cap D}$: The intersection between Enron$_T$ and Sheffield$_{all}$ in which disagree in labels.
- Enron$_{\cap}$: The intersection between Enron$_T$ and Sheffield$_{all}$.
- Enron$_{\cup}$: Sheffield$_{all}$ $\cup$ (Enron$_T$ $-$ Enron$_{\cap}$). In case of disagreement, we use Sheffield$_{all}$ labels.

3.4 Avocado Datasets

The Avocado Email Collection has 62,278 threads and 937,958 emails.

We have randomly sampled total of 2,000 threads and 5,339 emails from the Avocado corpus with different number of emails per thread as in Enron.

As described in Section 3.2, each annotator labeled 1,200 threads, with 400 threads in common. The first annotator has 3,197 emails, while the second has 3,207, and 1,065 emails are in common. After obtaining the final labels as described in Section 3.2, we got total of 1,976 threads and 5,280 emails labeled as either "Business" or "Personal" from the Avocado corpus. Table 2 shows the summary of the Avocado datasets with the following notations:

- Avocado$_1$: The threads and emails labeled by the first annotator as in 3.2.
- Avocado$_2$: The threads and emails labeled by the second annotator as in 3.2.
- Avocado$_{\cap A}$: The intersection between Avocado$_1$ and Avocado$_2$ in which both agree in labels.
- Avocado$_{\cap D}$: The intersection between Avocado$_1$ and Avocado$_2$ in which they disagree in labels.
- Avocado$_{\cap}$: The intersection between Avocado$_1$ and Avocado$_2$.
- Avocado$_{\cup}$: All the threads and emails labeled as in 3.2: Avocado$_1$ $\cup$ (Avocado$_2$ $-$ Avocado$_{\cap}$). In case of disagreement, we use Avocado$_1$ labels.

[2] we treat classes as completely different categories when computing Cohen's kappa

	Threads			Emails		
Set	Business	Personal	Total	Business	Personal	Total
Avocado_1	1,087 (91.2%)	105 (8.8%)	1,192	2,927 (92.1%)	251 (7.9%)	3,178
Avocado_2	1,035 (88.1%)	140 (11.9%)	1,175	2,851 (90.5%)	298 (9.5%)	3,149
$\text{Avocado}_{\cap A}$	340 (91.6%)	31 (8.4%)	371 (94.9%)	948 (93.3%)	68 (6.7%)	1,016 (97%)
$\text{Avocado}_{\cap D}$	NA	NA	20 (5.1%)	NA	NA	31 (3%)
$\text{Avocado}_{\cap}$	NA	NA	391	NA	NA	1,047
$\text{Avocado}_{\cup}$	340 (91.7%)	31 (8.4%)	1,976	4,826 (91.4%)	454 (8.6%)	5,280

Table 2: Summary of the Avocado datasets

3.5 Train, Development and Test Sets

For the binary classification task in this paper, only emails are used as data points. We defer the classification of threads to future work. We use three datasets for the experiments, namely: $\text{Enron}_{\cup}$, $\text{Enron}_{\cap A}$, and $\text{Avocado}_{\cup}$ (described in Section 3.3 and Section 3.4). $\text{Enron}_{\cup}$ and $\text{Enron}_{\cap A}$ are divided into into train, development and test sets with 50%, 25% and 25% of the emails respectively. $\text{Avocado}_{\cup}$ is divided equally into development and test sets (since we will not train on Avocado). For the rest of this paper, we refer to the train, development and test sets by subscripts tr, dev, and tes respectively.

4 Lexical and Local Features

For the classification task, we use pre-trained GloVe embedding vectors as lexical features (Pennington et al., 2014). There are various word vector sets available online, each trained from different corpora and embedded into various dimension sizes.

We use GloVe pre-trained word vector sets such that each email is represented by a vector of a fixed number of dimensions equal to the dimensionality of GloVe word vector set. We average all word vectors in the email using the pre-trained word vectors as follows:

$$e_j = \frac{\sum_i^n f_{e_j,v_i} v_i}{\sum_i^n f_{e_j,v_i}}$$

Here, f_{e_j,v_i} is the frequency of the word corresponding to vector v_i in email e_j, v_i is the word embedding vector in GloVe set. Both the body and subjects are included in the email content.

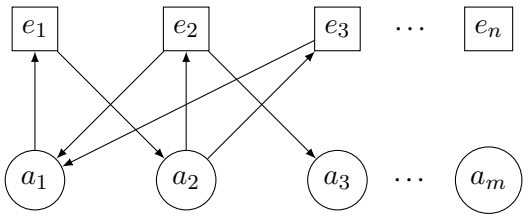
In addition to the contextual features, we use the number of recipient and the length of the email (in words) as meta-information that can be extracted from the email locally without looking at the email exchange network.

5 Social Network Features

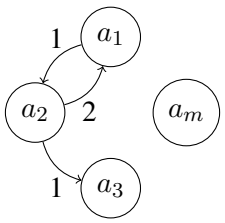
The email exchange network can be represented as social networks with different structures. One possible structure is to represent the email exchange network as a bipartite graph with two disjoint sets of nodes, emails and employees (i.e. email addresses) such that edges connect emails with employees, as edges between an email and employees exist if and only if their email address appears as either the sender or a recipient in that email; we refer to this structure as the email-centered network. Another structure is a graph (not necessarily bipartite) whose nodes represent employees (i.e. email addresses) and whose edges represent email communication such that and edge exists if there is a least one email has been exchanged between the two end nodes; we refer to this structure as the address-centered network. Figure 1 illustrates these two types of graphs. In both graphs we normalize multiple email addresses belonging to the same person into one email address (node).

For each corpus (i.e. Enron and Avocado), we construct directed and undirected graphs from these two networks (i.e. email-centered and address-centered). In directed graphs, each edge has a source and destination node, which shows explicitly the directionality of the email (i.e. sender and recipients), while in undirected graphs, the directionality of communication is not reflected within edges. In the case of the address-centered graph, the edge weight reflects the number of emails that have been exchanged between the two ends and the direction; in the case of the email-centered network, the weights are always 1. Different features from these types of graphs can be extracted.

We use the whole exchange network, including all labeled and unlabeled emails to build these graphs. We include features from both the sender and the recipients (either in the "to" or "cc" list). In case of the email has multiple recipients, we



(a) Email-centered network



(b) Address-centered network

Figure 1: Email Exchange graph

In-degree$^{\rightsquigarrow,\dagger,w,u}$	$indeg(v) = \sum_{u \in V} \mathrm{A}_{u,v}$ where: $\mathrm{A}_{u,v}$ is the weight of edge from u to v
Out-degree $^{\rightsquigarrow,\dagger,w,u}$	$outdeg(v) = \sum_{u \in V} \mathrm{A}_{v,u}$ where: Av, u is the weight of edge from v to u
Degree $^{\rightsquigarrow,\diamond,\dagger,w,u}$	$deg(v) = indeg(v) + outdeg(v)$
# common neighbors $^{\diamond,\dagger,u}$	$\lvert \bigcup_{r \in rec} \Gamma(s) \cap \Gamma(r) \rvert$ where: rec is the list of recipients s is the sender
# Sender's triangles $^{\diamond,\dagger,u}$	$\frac{1}{2} \sum_{v \in \Gamma(s)} \lvert \Gamma(s) \cap \Gamma(v) \rvert$ where: s is the sender
Jaccard's coefficient $^{\diamond,\dagger,u}$	$\frac{\lvert \bigcup_{r \in rec} \Gamma(s) \cap \Gamma(r) \rvert}{\lvert \bigcup_{r \in rec} \Gamma(s) \cup \Gamma(r) \rvert}$ where: rec is the list of recipients s is the sender
Fraction of triangles $^{\diamond,\dagger,u}$	$\frac{2 \lvert \bigcup_{r \in rec} \Gamma(s) \cap \Gamma(r) \rvert}{\sum_{v \in \Gamma(s)} \lvert \Gamma(s) \cap \Gamma(v) \rvert}$ where: rec is the list of recipients s is the sender
In-degree centrality $^{\rightsquigarrow,\dagger,w}$	$\frac{indeg(u)}{\lvert V \rvert - 1}$
Out-degree centrality $^{\rightsquigarrow,\dagger,w}$	$\frac{outdeg(u)}{\lvert V \rvert - 1}$
Degree centrality $^{\rightsquigarrow,\diamond,\dagger,w,u}$	$\frac{deg(u)}{\lvert V \rvert - 1}$
Betweenness centrality $^{\rightsquigarrow,\diamond,\dagger,\ddagger,w,u}$	$\sum_{s,t \in V} \frac{\sigma(s,t \lvert v)}{\sigma(s,t)}$ where: $\sigma(s,t)$ is the number of shortest paths between s and t $\sigma(s,t \lvert v)$ is the number of these paths that pass through v
Eigenvector centrality $^{\rightsquigarrow,\diamond,\dagger,\ddagger,u}$	For a node v: x_v where: x is the eigenvector corresponding to the largest eigenvalue of A $Ax = \lambda x$
Closeness centrality $^{\rightsquigarrow,\diamond,\ddagger,w,u}$	$\frac{\lvert V \rvert - 1}{\sum_{u \in V} d(v,u)}$ where: $d(v,u)$ is the shortest-path distance between v and u.
Auth Score $^{\rightsquigarrow,\diamond,\dagger,\ddagger,w,u}$	The authority score for a node using HITS algorithm (Kleinberg, 1999)
Hub Score $^{\rightsquigarrow,\diamond,\dagger,\ddagger,w,u}$	The hub score for a node using HITS algorithm.

$^{\rightsquigarrow}$ Extracted from directed graphs.
$^{\diamond}$ Extracted from undirected graphs.
† Features of senders/recipients in the Address-centered network.
‡ Features of emails in the Email-centered network.
^{w} Uses edge weights.
^{u} All edge weights are considered equal to 1.

Table 3: Social Network Features. A: the adjacency matrix for a graph (weighted or unweighted), $\Gamma(v)$: The set of neighbors of the node v

.

average the value corresponding to each feature. Table 3 summarizes the social network features.

6 Experiments

In this section, we present empirical results on the email classification task by conducting different

Classifier	Parameter	Parameter Space
SVM	γ	$10^{-4,-3,-2,-1,0}$
	kernel	rbf, linear
	C	$1, 10, 100, 1000$
Extra-Trees	# trees	$10, 20, 30, 50, 100, 200$
	Split Criteria	Gini, Entropy
	Min Sample	1, 3, 10
Both	Class-weights	{B:1, P:1}, {P:1, B:2} {P:1, B:3}, balanced

Table 4: Grid-search parameter space. B: Business, P: Personal. Balanced: class weights are adjusted inversely proportional to class frequencies in the training set

experiments on lexical and social network feature sets. We use three metrics to measure the performance, namely: accuracy score, Business F-1 score and Personal F-1 score. We are mainly interested in improving the Personal F-1 score since it is the minority class. We compare the performance of SVM classifiers and extremely randomized trees (commonly known as Extra-Trees) (Geurts et al., 2006) as implemented in the *scikit-learn* python library (Pedregosa et al., 2011). We tune the hyper-parameters using grid-search with 3-fold cross-validation on the training set. Table 4 shows the grid-search space for the two classifiers. As a preprocessing step, we apply logarithmic transformation on the network and meta-information feature values to be approximately normal in distribution. Then, all feature values (i.e. lexical, network and meta-info) are standardized to have zero-mean and unit-variance.

Vector Set	Accuracy (%)	F-1 B (%)	F-1 P (%)
BOW	92.3	95.6	71.2
6B.50d	93.0	95.9	75.7
6B.100d	93.0	95.9	75.5
6B.200d	95.0	97.1	80.0
27B.25d	94.5	96.8	80.0
27B.50d	94.3	96.7	79.2
27B.100d	95.0	97.1	80.7
27B.200d	93.7	96.3	77.6
42B.300d	**95.4**	**97.3**	**83.1**
840B.300d	95.1	97.2	80.5

Table 5: Results from different GloVe word vector sets and a BOW model as a baseline trained on $\text{Enron}_{\cap A\ tr}$ and tested on $\text{Enron}_{\cap A\ dev}$.

6.1 Obtaining Best GloVe Vector Set

First, in order to obtain the GloVe vector set that maximizes the performance, we experiment with different GloVe pre-trained vectors as lexical features (meta-information features are not included). Table 5 shows the results of classification of different GloVe pre-trained vector sets trained on $\text{Enron}_{\cap A\ tr}$ and tested on $\text{Enron}_{\cap A\ dev}$. In addition, a bag-of-words (BOW) model is shown as a baseline. In this model, we represent each email as a vector of frequencies (term counts), then we select the top 500 words using χ^2 feature selection method. In all models (i.e. GloVe vectors and BOW), we use SVM classifiers and we tune parameters using grid-search.

The results show that, in general, more training data is better, and more dimensions are better. However, the best set is the 300-dimensional 42B.300d which is trained on a large 42 billion token corpus, rather than the larger 840 B words-based embeddings. We use these embeddings in all further experiments.

6.2 Experiments with Different Features and Sets

In this subsection, we perform experiments with different models tested on $\text{Enron}_{\cup\ dev}$ and $\text{Avocado}_{\cup\ dev}$. We assume that the ultimate application of our work is a setting in which we train models on a company (i.e. Enron) and apply it to another company (i.e. Avocado).

First, we tune the hyper-parameters using grid-search with 3-fold cross-validation on $\text{Enron}_{\cup\ tr}$ and $\text{Enron}_{\cap A\ tr}$ three times: first, using network and meta-information features only, second, using lexical (embedding) features only, third, using all features.

Then, we select the best SVM and Extra-trees models with the lexical features only and the models with all features. We apply a paired t-test on the personal F-1 scores of of the models (i.e. SVM and Extra-trees models with lexical features only and with all features) using 10-fold cross-validation.

The results of the paired t-test show that the improvement obtained from adding the network features is statistically significant on $\text{Enron}_{\cup\ tr}$ ($p < 0.05$), but not on $\text{Enron}_{\cap A\ tr}$ ($p > 0.05$) using both SVM and Extra-trees classifiers.

For evaluating how well the models will perform in an intra-corpus setting, we test on $\text{Enron}_{\cup dev}$, using models trained on $\text{Enron}_{\cup tr}$ with different classifiers and feature sets. Table 6 summarizes the intra-corpus results. These results

				Business			Personal		
Trained on	Classifier	features	Accuracy	F-1	Recall	Precision	F-1	Recall	Precision
Enron$_{\cup\ tr}$	SVM	Net	83.6	89.4	87.2	91.7	64.0	70.0	58.9
		Lexical	90.2	93.8	92.4	95.1	77.7	81.9	73.9
		All *	90.0	93.5	91.1	**96.1**	**78.1**	**85.9**	71.7
	Extra-Trees	Net	87.2	92.0	92.9	91.2	68.1	65.7	70.6
		Lexical	88.9	93.1	95.3	91.0	70.5	64.2	78.3
		All	**91.3**	**94.7**	**97.1**	92.4	76.9	69.4	**86.1**

Table 6: Results of different classifiers tested on Enron$_{\cup\ dev}$. Net features include meta-information features

				Business			Personal		
Trained on	Classifier	features	Accuracy	F-1	Recall	Precision	F-1	Recall	Precision
Enron$_{\cup\ tr}$	SVM	Net	85.7	92.1	89.9	94.3	26.7	34.3	21.9
		Lexical	89.2	93.9	89.9	**98.2**	53.0	**80.1**	39.6
		All	90.2	94.5	91.7	97.5	52.6	71.6	41.5
	Extra-Trees	Net	91.1	95.3	97.6	93.1	17.5	12.4	29.8
		Lexical	92.0	95.7	94.8	96.5	52.9	58.7	48.2
		All	92.3	95.8	95.6	96.1	51.2	52.7	49.8
Enron$_{\cap A\ tr}$	SVM	Net	89.2	95.8	95.6	96.1	51.2	52.7	49.8
		Lexical	94.3	96.9	97.3	96.5	60.7	57.7	64.1
		All *	**95.0**	**97.3**	98.2	96.5	**63.0**	56.2	71.5
	Extra-Trees	Net	92.0	95.9	**99.5**	92.5	3.7	2.0	23.5
		Lexical	93.7	96.7	98.9	94.6	43.2	31.3	69.2
		All	93.8	96.7	99.0	94.6	43.2	30.8	**72.1**

Table 7: Results of different classifiers tested on Avocado$_{\cup\ dev}$. Net features include meta-information features

			Business			Personal		
Trained on	Tested on	Accuracy	F-1	Recall	Precision	F-1	Recall	Precision
Enron$_{\cup\ tr}$	Enron$_{\cup\ ts}$	91.2	94.4	92.1	96.7	79.9	87.5	73.5
Enron$_{\cap A\ tr}$	Avocado$_{\cup\ ts}$	93.5	96.4	96.9	96.0	64.7	62.1	67.7

Table 8: Applying best models on test sets. Both models are SVM classifiers trained with all features.

show that adding network features helps in retrieving more personal emails (increasing the personal recall) when using both classifiers. In addition, it is clear from the results that the network features are more effective with Extra-Trees since adding them improves all the scores.

To evaluate the cross-corpora performance, we test on Avocado$_{\cup\ dev}$ using different models trained on Enron$_{\cup\ tr}$ and Enron$_{\cap A\ tr}$. Table 7 summarizes the cross-corpora results. We use Enron$_{\cap\ Atr}$ in this experiment to test how well a model performs on another corpus when training on a dataset with few but high-confidence labels, in comparison with training on a larger dataset with labels of lesser confidence. The results show that a model trained on a large dataset with lesser confidence labels (i.e. Enron$_{\cup\ tr}$) using lexical feature alone can retrieve many personal emails, but with a poor precision. Unlike the intra-corpus setting, adding network features always increases the personal precision but decreases the personal recall. However, the best performance as measured by f-measure is achieved by combining the network and lexical features, and using SVMs, which is the same best configuration as in the intra-corpus evaluation setting. For the inter-corpora evaluation, the best result is achieved using the smaller training corpus with higher quality labels.

In both settings (i.e. intra-corpus and cross-corpora), Extra-Trees classifiers suffer in retrieving personal emails causing a decrease in the F-1 personal score in comparison with SVM classifiers.

6.3 Performance on the test set

Finally, we select the models with the highest F-1 score each both task (intra-corpus and cross-corpora), and then we test these models on Enron$_{\cup\ ts}$ and Avocado$_{\cup\ ts}$. Table 8 shows the per-

formance of the best models on the test sets. The results show that in an intra-corpus setting, we can achieve a high personal F-1 score. Also, it is possible to get a good performance on a corpus (i.e. Avocado) when training on another one (Enron).

7 Conclusion and Future Work

In this paper, we have shown that classifying emails into business and personal can be predicted with good performance using conventional classifiers trained with pre-trained word embeddings that are available online. We performed different experiments on two corpora, Enron and Avocado. The cross-corpora results show that it is possible to classify emails of a company using models trained on another company with a good performance. In addition, we have shown that including features obtained from the graphs representing the email exchange network improves the classification performance.

We observe that the percentage of personal email decreases from 20% (in Enron) to less than 10% (in Avocado). It is not clear whether this is due to the nature of two companies or due to the spread of free email services such as Hotmail and Gmail.

In the future, we plan to experiment with adding more network features that can capture more global network features using approaches such as graph spectral analysis and graph kernels.

8 Acknowledgements

We would like to thank Ibrahim Almosallam for helpful discussions. We would like to thank the anonymous reviewers for their valuable feedback. The first author was supported by the KACST Graduate Studies program. The second author was supported for this work in part by the DARPA DEFT program. The views expressed here are those of the authors and do not reflect the official policy or position of the U.S. Department of Defense or the U.S. Government.

References

Apoorv Agarwal, Adinoyi Omuya, Aaron Harnly, and Owen Rambow. 2012. A comprehensive gold standard for the enron organizational hierarchy. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*. Association for Computational Linguistics, pages 161–165.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning* 63(1):3–42.

David Graus, David Van Dijk, Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. 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*. ACM, pages 1079–1082.

Johanna Hardin, Ghassan Sarkis, and PC Urc. 2014. Network analysis with the enron email corpus. *arXiv preprint arXiv:1410.2759* .

Sanaz Jabbari, Ben Allison, David Guthrie, and Louise Guthrie. 2006. Towards the Orwellian nightmare: separation of business and personal emails. In *Proceedings of the COLING/ACL on Main conference poster sessions*. Association for Computational Linguistics, pages 407–411.

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

Jon M Kleinberg. 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* 46(5):604–632.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *Machine learning: ECML 2004*, Springer, pages 217–226.

Tanushree Mitra and Eric Gilbert. 2013. Analyzing gossip in workplace email. *ACM SIGWEB Newsletter Winter* 5.

Douglas Oard, William Webber, David Kirsch, and Sergey Golitsynskiy. 2015. Avocado research email collection. *Philadelphia: Linguistic Data Consortium* .

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12(Oct):2825–2830.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.