Data Mining Academic Emails to Model Employee Behaviors and Analyze Organizational Structure

Kayla M. Straub

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Robert W. McGwier, Chair

Aloysius A. Beex

R. Michael Buehrer

Bert Huang

March 25, 2016

Blacksburg, Virginia

Keywords:

Data Mining Academic Emails to Model Employee Behaviors and Analyze Organizational Structure

Kayla M. Straub

(ABSTRACT)

Email correspondence has become the predominant method of communication for businesses. If not for the inherent privacy concerns, this electronically searchable data could be used to better understand how employees interact. For example, after the Enron dataset was made available, researchers were able to provide great insight into employee behaviors based on the available data despite the many challenges with that dataset. The work in this paper demonstrates the application of a suite of methods to an appropriately anonymized academic email dataset created from volunteers' email metadata. This new dataset, from an internal email server, is first used to validate machine learning and feature extraction algorithms and then to generate insight into the interactions within the center. Based solely on email metadata, a random forest approach modeled behavior patterns and accurately classified employees by job title. The algorithm performed very well not only on participants in the study but also on other members of the center who were connected to participants through email. Furthermore, the data revealed relationships not present in the formal operating structure. The result is a much fuller understanding of the center's internal structure than can be found in the official organization chart.

Acknowledgments

I want to acknowledge the many people who helped me with my graduate work at Virginia Tech and made this work possible. Thank you to the members of my committee: Dr. Robert W. McGwier, Dr. Louis Beex, Dr. Michael Buehrer, and Dr. Bert Huang for your guidance. In particular, I would like to sincerely thank Dr. McGwier for his vision and support throughout this research.

I want to express my gratitude to the faculty, staff, and students of the Hume Center for all of your help and for providing me with the opportunities to work on such interesting research. I am especially thankful for Dr. Joseph Ernst and Dr. William C. Headley for their invaluable insight and assistance.

I would also like to thank my family and friends for their support throughout my college career. I want to specifically thank my husband Ben for his unwavering support and for always pushing me to be my best.

Finally, I thank God for all the blessings in my life and for getting me to where I am today.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

A reorganization of a business can be very costly and has far-reaching effects once implemented, either for better or for worse. While this official hierarchy is important, there is an equally important organic organization of any business, which may or may not be reflected in the official organization chart. Understanding this unofficial structure could be invaluable during such a business reorganiztion, but due to its informal nature, it can be difficult to determine. One massive source of electronically searchable information that could be used to better understand this hidden structure is the business's emails. However, privacy concerns inhibit most research thrusts into email analysis.

Extracting hierarchical relationships from email metadata can be used for several potential applications beyond gaining a better understanding of the organizational structure. For example, it could be possible to identify people of influence or potential leaders. By recognizing changing behavior, potential insider threats could be identified.

1.2   Problem Overview

This thesis addresses the problem of automatic organizational determination. Specifically, it considers using email analysis to perform this operation. However, there is a lack of modern, publicly available email datasets to use for this research. Out of the email datasets that do exist, namely the Enron email corpus, there is a lack of accurate job title labels for the employees. Labels are necessary for evaluating the results of the organic hierarchy analysis with the official organization chart. A new source of workplace emails with accurate job title labels was required.

The research presented in this thesis documents how a new email metadata dataset was collected and appropriately anonymized from the email records of 37 voluntary participants from an academic research organization. This metadata is used not only to analyze the participants, but also to what extent the non-participant members of the organization can be characterized.

The problem of analysis is broken into two parts. The first half addresses accurately classifying the employees of the center by their job title. This will give insight into how well email behaviors

indicate the official hierarchy as well as how similar the job titles are to one another. The second part of the problem investigates how well this organic organizational chart matches the official organizational chart.

1.3   Approach

These emails are used to calculate a high-dimensional feature set to be used with machine learning techniques. The 114 features used in this study can be grouped into two common areas in email analytics: traffic-based and social-based. Using random forests, these features are used to accurately predict each employee's job title. Finally, a comparison is presented between relationships displayed in the data with the formal organizational chart. Combining predicted job titles with hierarchical relationships discovered from the data can identify an organization's true structure.

1.4   Contributions

The contributions outlined in this thesis are broad. First, a new, fully anonymized dataset was developed from raw emails from an academic research environment. This dataset, unlike others before it, has accurate job title labels. Furthermore, a unique combination of features were calculated from this data. Some had been used in email analysis before, but many are new and specific to this data. Research investigating this problem in the past has focused on one of the two types

of features, where this analysis aggregates traffic- and social-based features together. In general, the problem of job title classification has not been extensively studied. The results in this thesis surpass the results of previous research. Finally, a closer look at email relationships lends insight into the organic structure of the organization.

1.5 Outline

This thesis continues by discussing the related works in Section 2. Section 3 describes the process of data collection and some statistics of the dataset and the features extracted from the data. The methods investigated using these features are covered in Section 4. The results of the analysis are presented in Section 5. Section 6 presents opportunities for future work, and Section 7 concludes the thesis.

Chapter 2

Prior Work

This chapter investigates the previous research performed in the area of email analysis. Context is established with a discussion on the importance of workplace email communication in modern society. A description of the benchmark in email datasets, the Enron email corpus, is provided, and the drawbacks of this dataset are highlighted. The last section provides an overview of features typically used in email analysis and the works that address the problem of organization determination.

2.1 Background

Email is a pervasive medium for communication in modern society—particularly in the workplace. In 2015, there were over 2.6 billion email users [1]. It is projected that by the end of 2019, over one third of the global population will be using email. In fact, the average business email user sends and receives a total of 112 emails per day. Corporate email alone accounts for 54.7% of worldwide email traffic. Retention of large email archives has become common practice with decreasing memory size and cost [2]. Out of the 600 employees involved in that study at Microsoft, the average employee had 28,660 emails stored in 133 folders which is a significant increase over the past ten years. This is a considerable amount of untapped information that could be leveraged to characterize employee roles within an organization.

2.2 Commonly Used Datasets

When the Enron email corpus was released in 2004 [3], it became the one of the largest email datasets available for public use. It is unique in that it included all of the email text, which lead to many studies involving text analysis and natural language processing. Since then, this dataset has been extensively researched on topics including spam classification [4], [5]; email categorization [6], [7]; and recipient prediction [8], [9]. However, there are known flaws and discrepancies with even the most recent versions of this dataset—ranging from misspelled email addresses [10] to

duplicate email addresses [11], and misfiled emails [12]. In one of the most popular forms of the dataset [13], the database includes 253,735 emails sent as "CC" and 253,713 emails sent as "BCC". Further inspection reveals that emails sent as one type or the other were almost always mistakenly recorded as both.

There exists a list of ground truth job title labels for the Enron email corpus, enabling corporate hierarchy analysis. The research in [14] made use of these labels along with performing natural language processing functions on the email text. This work identified phrases more likely to be used when emailing superiors compared to phrases used when emailing peers or subordinates. In [15], the corpus and job title labels were used to explore pattens in office email gossip with respect to the organizational hierarchy. There are clear issues with the job title labels, though. For example, Jeff Dasovic is labeled as an Employee, seemingly the lowest title. However, he had more emails than anyone else in the dataset and is identified in records as Director for State Government Affairs. As noted in [14], there are similar inconsistencies for Sally Beck, Rick Buy, and Rod Hayslett. Amidst all of these concerns about the Enron job title labels, this work proposes using a new dataset with accurate job titles generated from intimate knowledge of the organization.

2.3   Prior Analysis

The existing literature on analyzing social email behavior is mainly divided into two categories: traffic-based and social-based [16]. Traffic-based methods calculate statistics based only on email

patterns. Social-based methods represent the email communications as a social graph and then extract information from this model using graph-based algorithms.

Using features extracted from email metadata, [17] was able to cluster levels of management at Enron. In addition to email traffic statistics, using features such as the presence of different email attachment types and the length of emails was shown to successfully categorize email behavior in [18].

Relational ties can be modeled as a graph network where nodes represent people and edges represent email interactions. This is a useful model because many statistics can be calculated from the layout of a social graph [19]. A common metric that has been shown to indicate importance in a social graph is betweenness centrality, which comes in several different flavors first developed by [20]. Betweenness centrality is a measure of how many shortest paths in a graph travel over each node. A node with high betweenness centrality in a social graph has been shown to represent a high degree of influence on other nodes. As shown in [21], a betweenness centrality algorithm can be used to determine community structures within an organization. There are several other types of centrality such as closeness, current-flow betweenness, and random-walk centrality [22]. In [23], researchers were able to more accurately predict superior-subordinate pairs within Enron using only degree centrality than some natural language processing methods. Other successful metrics were used in [24], which detected the most important email users within a corporate network without using betweenness or centrality as features. The features used in that study were: degree, the number of edges connected to a node; density, the ratio of actual edges to the number

of possible edges; and proximity prestige, the ratio of nodes that can reach a node i to the average distance from those nodes to i.

Instead of considering only traffic-based or social-based analytics, these can be be used jointly. An example of this approach [25], which combined features such as number of emails, response time, cliques, and degree centrality into a "Social Score", was used to rank Enron employees.

Chapter 3

Email Dataset and Feature Extraction

This chapter details all of the steps taken to transform the raw data into the high-dimensional feature space used in the learning algorithm. Information is provided on the data collected and statistics about the dataset. There is a detailed description of all features used, both traffic-based and social-based.

3.1 Data Collection

Over the past decade, the Enron dataset has been widely used to study email behaviors because it is one of the only datasets available comprised of real-world corporate emails. A list of ground truth job titles was compiled by [26]. However, there are known issues with these labels. Due to

difficulties with the Enron dataset, as described in Section 2, this study uses a new dataset generated from volunteers in one of the university's centers.

Due to the inherent privacy concerns, it was necessary to work with the Internal Review Board (IRB) to approve a dataset which maintains participants' privacy. This dataset is meant to be representative of metadata which any company could use without violating the privacy of their employees. Special care was taken to protect the privacy of those involved in the study. During the collection process, all subject and body text was hashed, and all email metadata was stored in a MySQL database using scripts without any researchers observing any email text. Furthermore, any identifying information has been omitted from this thesis.

Most of the collected data is information from the header of an email such as the sending and receiving email addresses and date and time. The number of characters in the subject and body text were extracted before the subject and body text was hashed. The prefix of the email was also recorded before hashing. The hashed body and subject were loaded into the database in order to discern between emails with the same text. For example, if an email is forwarded, it will have the same hashed subject once the prefix is removed, and it is therefore possible to count the number of times an email with the same subject has been received. The number of attachments to each email was also collected. The center's email server offers the ability to both digitally sign and encrypt emails. Sending an email with either or both of these options appears as a special type of attachment in the raw email file. Therefore, during the data collection process, information about if an email was signed or encrypted was recorded. All of the metadata was collected by a

Table 3.1: The list of all information collected from each email within the study.

Destination and source email address
Email time stamp
Subject prefix (e.g., Re:, Fwd:)
Hash of subject after removing prefix
Hash of body text
Length of subject in characters
Length of body text in characters
Number of attachments
Indicator if email was digitally signed
Indicator if email was encrypted

comprehensive parser script. The types of information extracted from each email is summarized in Table 3.1.

3.2   Dataset Description and Statistics

Table 3.2 compares statistics between this internal dataset and the Enron corpus. This internal database is more modern, contains more emails, and covers a longer time period. However, it is comprised of fewer people than the Enron dataset.

Table 3.2: A comparison between the internal dataset and the Enron email corpus.

	Center	Enron
Time	11/2012-11/2015	1/2000-9/2002
Distinct Email Addresses	32,118	75,406
Participants	37	249
Distinct Emails	585,096	252,759

The center divides its employees into six main areas: directors, graduate students, operations, outreach, project management (PM), and research. A chart showing the distribution of classes is shown in Figure 3.1.

While the study includes only 37 volunteers, the email metadata from these volunteers identified 32 additional employees of the center. These additional employees were included in the study when ground truth for their job was available and when they were identified in at least 100 email metadata records. This provides 69 total employees in the study.

3.3   Features

The study includes 114 features that were extracted from the email data: 84 traffic-based and 30 social-based. In the following sections, all features from each of the two categories are described.
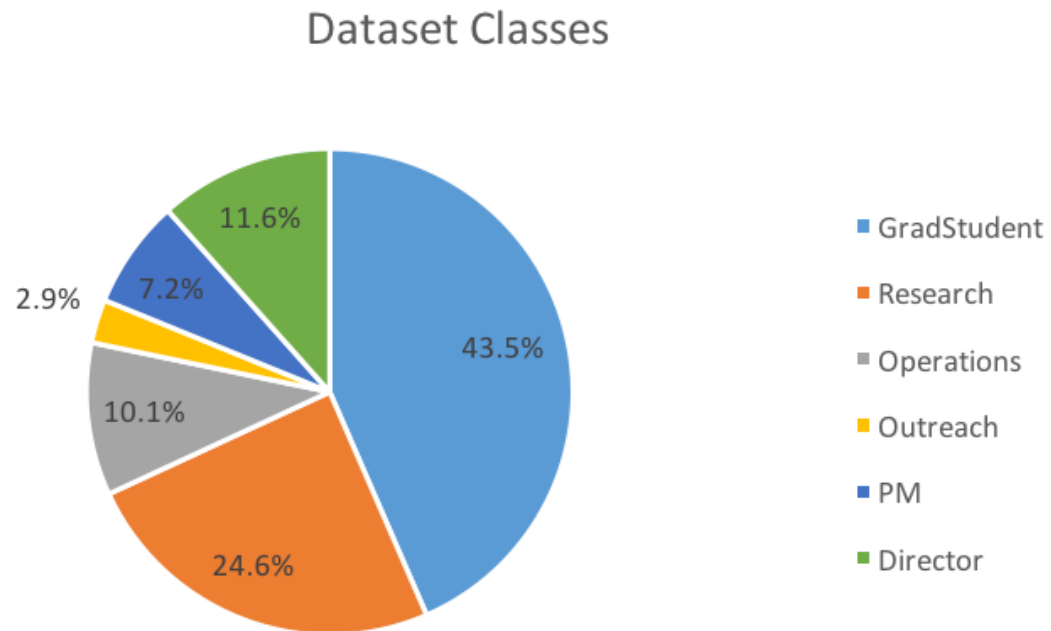
Figure 3.1: Pie chart showing the distribution of classes in the dataset. Note that the classes are far from being uniformly distributed.

3.3.1 Traffic-Based Features

The traffic-based features are those calculated purely from the collected email metadata. This section details all of the traffic-based features used in this analysis and how they are calculated.

Email Counts and Email Types

The simplest traffic-based features involve counting how many and what kinds of emails each participant sends and receives. First, the total number of emails, total sent, and total received give a measure of how active an email user is on average. These features can also indicate the direction

tendencies of an employee's communication. Do they send more emails than they receive, or vice versa? All of the traffic-based features detailed below consider direction. Specifically, each metric is calculated three times: considering only sent emails, only received emails, and both sent and received emails.

Some traffic-based features focus on the different types of emails. Two examples of these types of features are the number of emails sent directly to each employee and the number of emails where they were carbon copied on the email. The opposite direction of this was inspected as well, that is, the number of emails the employees sent directly to others and the number of carbon copies sent out. The average number of recipients on emails sent and received for each participant were also calculated. Similarly, the number of emails sent and received as replies or forwards were used. These measures give a sense of how the employee communicates with others in the organization and their connectedness.

Several features were calculated from just the subset of emails that were digitally signed. These features were the total number of emails sent and received, number of unique email addresses, and the number of unique subjects. These same metrics were also calculated for encrypted emails.

Metadata Statistics

The metadata of the emails contains extremely useful information. This includes the email addresses involved, time stamp, the subject and body hashes and character counts, and the presence

of any attachments. From the time stamp, the time of day for each email was available. The total number of emails with timestamps after hours were used as a metric. For this purpose, after hours was defined as between 6pm and 7am EST on weekdays or anytime on weekends. The timestamps were also used to calculate the average number of emails per day for each employee. The mean and variance of the number of characters in the subject and body were calculated. The total number of attachments sent and received were computed as well as the average number of attachments per email.

Some of the most interesting information came from email addresses and subject hashes of the emails. The number of unique email address connections, both sent and received, was used as a feature. By counting the number of identical hashed subjects, it was determined how many unique subjects were both sent and received from each employee. The motivation behind these features is that employees with particular job titles may be more likely to be associated with long email chains, which would have the same subject. It was hypothesized that staff members had more external communications than graduate students. To test this, the number of emails sent and received from within the center and the university were calculated. Email addresses with a Virginia Tech domain were considered to be affiliated with the university. Email addresses with accounts on the internal mail server were labeled to be within the center. Note that all employee email addresses of the center have a Virginia Tech domain, and are therefore also considered to be part of the university.

Most of the features described above involved raw email counts. However, this could skew data by

giving more importance to employees who have been associated with the center longer. In order to normalize these values, corresponding percentage values were also fed into the learning algorithm. Examples include the percentage of sent emails that were sent after hours and the percentage of received emails with unique subjects out of all received emails.

Most Useful Traffic-Based Features

The best traffic-based feature for predicting employee status was the number of unique subjects received. The metric used to evaluate features is information gain. This was used because it is integral to the machine learning algorithm described in Section 4; details on the feature ranking analysis are provided in that section. This number of unique subjects received represents how many distinct conversations involve each individual. It is intuitive that people who are involved in more conversations hold a higher position in an organization because their input is requested more often. A histogram showing the different values for this metric over the different job classes is shown in Figure 3.2. This figure demonstrates how difficult this classification problem can be. It is clear from the figure that graduate students participated in far fewer email conversations than any other group. However, at least one graduate student is in six out of the first seven bins. Another issue is that there are different personalities among the staff, which might explain why there are at least two research employees and at least one director in the first bin (< 500) and last bin (> 5000). The divisions among the classes become slightly more clear when the feature values are normalized between 0 and 1, as opposed to the raw numbers shown here. All feature values are normalized
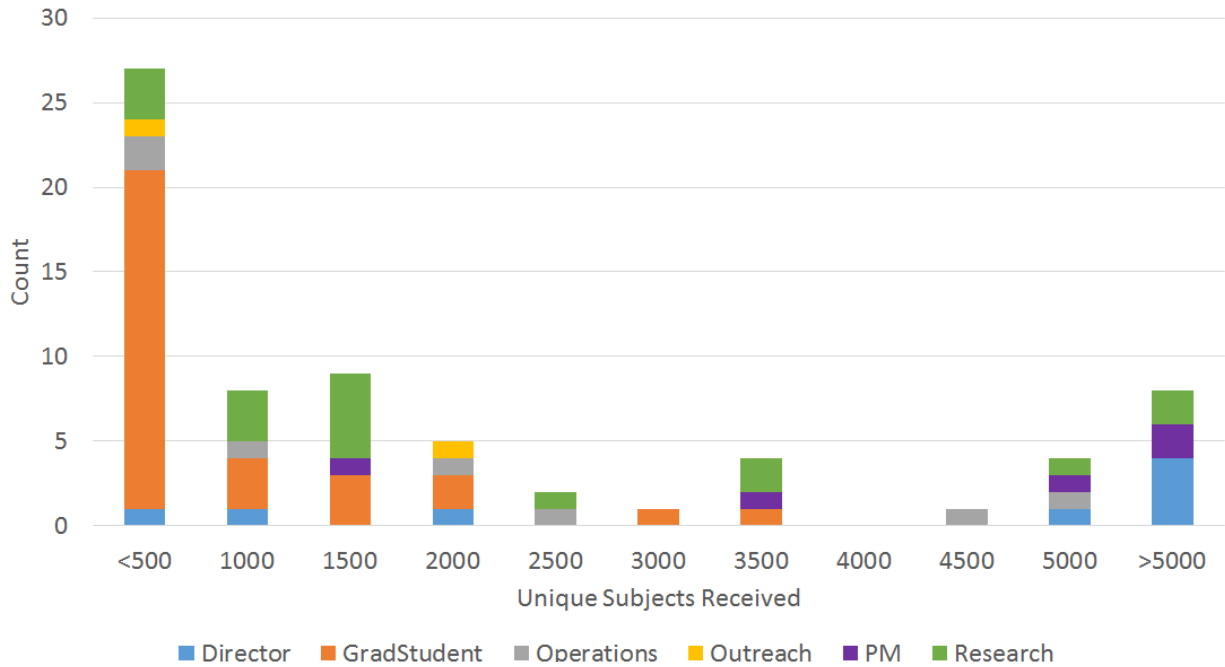
Figure 3.2: Histogram of unique subjects received by job title. Note that by using different thresholds, meaningful splits in the data can be made. For example, employees with less than 500 unique subjects received are likely to be graduate students, and employees with greater than 4500 are likely to be higher status employees (i.e., Directors, PMs, some Research).

before being used by the learning algorithm.

The second best traffic feature was the number of signed emails received. Signed emails usually signal sensitive information. Only certain groups within the center deal with this type of information, therefore it is understandable that this feature could help divide the subjects by title. Finally, the third best traffic feature was the number of emails received as forwards. Typically, those higher in the chain of command are forwarded emails where graduate students and lower-level employees are more likely to receive either replies or emails sent directly to them. Notice that there are

intuitive explanations behind all of the features selected by the ranker.

3.3.2 Social Network Features

In addition to tracking metadata statistics, features are also derived from modeling the emails as a social network. The social features first transform email patterns in a graphical network and then calculate statistics from this model.

Social Network Representation

A social network is composed of nodes, which represent people, and edges, which represent the emails between people. For this analysis, two different graphs were generated for analysis. In the full graph, an edge exists between any two individuals that exchanged at least one email. Each edge was given a weight equal to the total number of emails exchanged between the two employees. The edges are undirected for this analysis. A second graph only produces the same weighted edge between two nodes but only if at least 10 emails were exchanged. The purpose of this second graph is to filter out stray single-email relationships between coworkers that do not constitute meaningful communication. The full graph including all of the center employees is shown in Figure 3.3. Note that graduate students are found on the fringes of the network and are generally the least connected. However, there is a clear core group of employees that communicate very frequently.
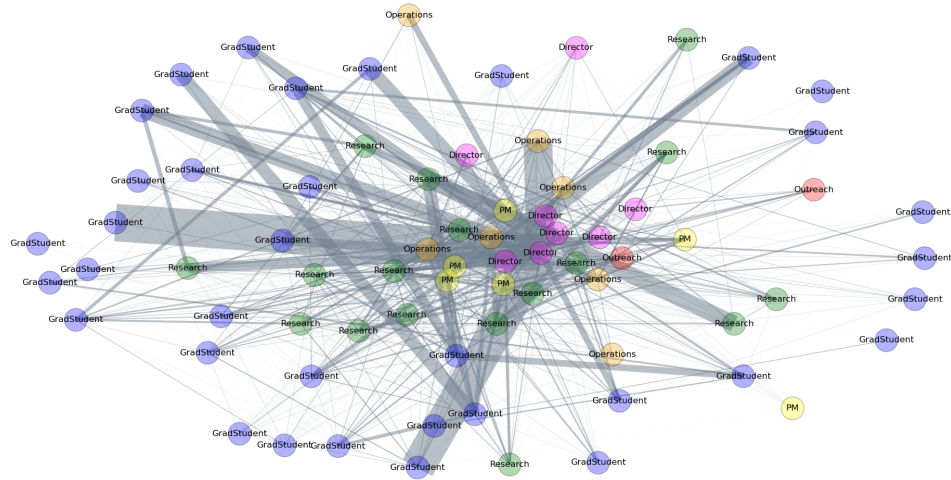
Figure 3.3: A social graph representation of the center. Nodes represent employees, and the thickness of the edges between nodes represent how many emails were exchanged.

Another representation of the full graph is shown as an adjacency matrix in Figure 3.4. Each of the two axes represent the employees of the center. The color at each coordinate indicates how much communication existed between the two employees. Some employees never exchanged any emails, while others exchanged many.

Using this model, a suite of statistics can be calculated about the people in the graph. The graphs were modeled in python. The graph library networkx [27] was used to calculate all graph features.
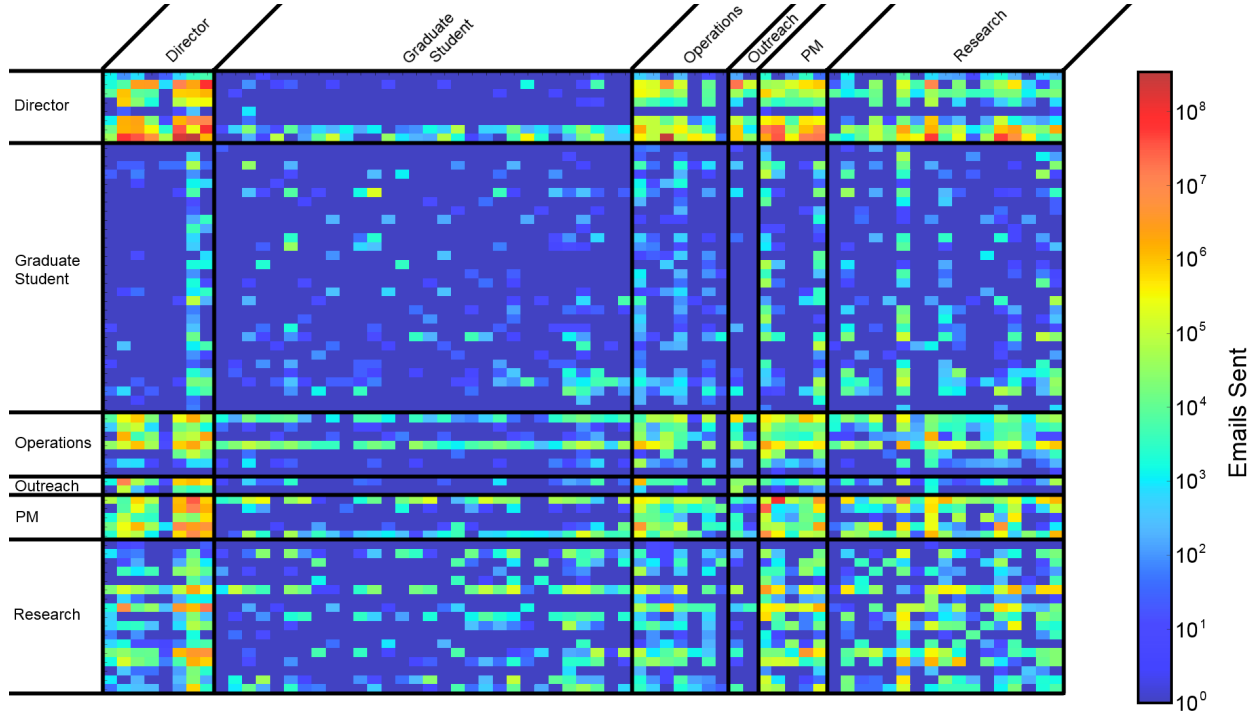
Figure 3.4: The adjacency matrix representing the social connections of the center. This graph overall is well connected and has just one component. Nonetheless, there are many pairs of individuals who never exchanged a single email.

Degree Measures

The degree of each node, from both the full and partial graph, was used as a feature. The degree of a node i is simply the number of other edges connected to node i. All graph features were calculated once for the full graph and again for the partial graph. The average neighbor degree was used as another feature. The neighborhood of node i is comprised of all nodes that are connected to i via edges. Therefore for node i, this metric averages the degree of each node in the neighborhood

of i. Mathematically, this is:

$$k_{\text{avg},i} = \frac{1}{|N(i)|} \sum_{j \in N(i)} k_j \qquad (3.1)$$

where $N(i)$ are the neighbors of node i and k_j is the degree of node j. The distance between nodes

were also used to generate some features. In graph theory, distance is measured by the length of the

path between two nodes. Between node i and any other node j in graph $\mathcal{G}$, there exists a shortest

path, $d(i, j)$. The average shortest path between node i and all other nodes in the graph, $d_{avg,i}$, was

used as a metric for the learning algorithm. That is,

$$d_{avg,i} = \frac{1}{n-1} \sum_{j \in \mathcal{V}, j \neq i} d(i, j) \qquad (3.2)$$

where n is the number of nodes in graph $\mathcal{G}$ and $\mathcal{V}$ is the set of nodes in $\mathcal{G}$. Similarly, the maximum

shortest path length, or eccentricity, was used as a feature in the learning algorithm. All of these

measures can be interpreted to represent the centrality of a node. If a node has many neighbors

with large degrees or if it has very short maximum shortest paths it is probably representative of a

person well-connected within the center.

Cliques

Some of the social features were based on existing graph theory concepts and algorithms. For

example, cliques. If a subgraph of a graph $\mathcal{G}$ is maximally connected, that is all nodes are connected

directly to each other, then this is called a maximal clique. The number of cliques to which a

node belongs was used as a feature. The motivation behind using this metric is that it should

mirror working groups within the center. Therefore, the more groups an employee belongs to or communicates with, the more important they are assumed to be.

Adapting Search Engine Algorithms

The hubs and authorities of each node in both graphs were calculated. The terms hubs and authorities come from the Hyperlink-Induced Topic Search (HITS algorithm) developed by [28]. This algorithm was originally designed to rate web pages, but has since been applied to social networks. A node's authority is just that—a measure of its importance over other nodes. A node's hub score is a measure of how well-connected it is to other nodes.

Another algorithm used to generate features was the pagerank algorithm, developed by Google [29] to rank webpages for search results. The assumption is that the most important webpages will be linked to frequently by other pages. Therefore, the ranking is determined by estimating the quality and quantity of links to a node. Both of these algorithms were shown to predict expertise within an online social network in [30].

Clustering Metrics

The triangle clustering coefficient, C_3 was also used as a metric. Consider a node i with neighbors m and n. This value, developed by [31], measures the probability that m and n are also connected.

This is calculated by comparing the number of triangles within the graph to the maximum number of possible triangles in the graph. If node i has degree k_i, there can be at most $\frac{k_i(k_i-1)}{2}$ triangles formed in this subgraph. Recall that the social networks are weighted graphs. The weights of the graph are incorporated to this metric by finding the geometric mean [30]. Therefore, triangle clustering coefficient for node i, $C_{3,i}$, is:

$$C_{3,i} = \frac{2}{k_i(k_i-1)} \sum_{m,n} (\tilde{w}_{i,m}\tilde{w}_{m,n}\tilde{w}_{n,i})^{\frac{1}{3}} \tag{3.3}$$

The edge weights in this calculation must be normalized compared to the maximum weight in the subgraph, i.e. $\tilde{w}_{i,m} = \frac{w_{i,m}}{max(w_{i,m})}$.

The square clustering coefficient is very similar and was also used in the algorithm. This metric, developed by [32], measures the probability that m and n are also neighbors to a fourth node, p. This configuration would form a square. To simplify the calculations, the graph for this metric is viewed without edge weights. Therefore, the square clustering coefficient for node i, $C_{4,i}$, is the proportion of actual squares within a subgraph centered around node i to the maximum number of possible squares in the same subgraph. This is calculated as:

$$C_{4,i} = \frac{\sum_{m=1}^{k_i} \sum_{n=m+1}^{k_i} q_i(m,n)}{\sum_{m=1}^{k_i} \sum_{n=m+1}^{k_i} [a_i(m,n) + q_i(m,n)]} \tag{3.4}$$

where $q_i(m,n)$ is the number of neighbors shared by m and n, excluding i and

$$a_i(m,n) = (k_m - \eta_i(m,n))(k_n - \eta_i(m_n)) \tag{3.5}$$

where $\eta_i(m,n) = 1 + q_i(m,n) + \theta_{mn}$. It is defined that θ_{mn} is an indicator function that takes on a value of 1 if m and n are neighbors and 0 otherwise. In theory, the higher the clustering coefficient, the more connected the node is within its neighborhood.

Centrality Measures

The majority of the social-based features were variations on centrality measures. First is closeness

centrality. Closeness centrality, $C(i)$ is the normalized inverse of the sum of shortest path distances

from node i to all other nodes in the graph [33]. It is calculated as follows:

$$C(i) = \frac{n-1}{\sum_{j=1}^{n-1} d(i,j)} \tag{3.6}$$

where n is the number of nodes in graph $\mathcal{G}$ and $d(i,j)$ is the minimum shortest path distance

between node i and node j. Since for this application $\mathcal{G}$ is a weighted graph, shortest path distances

are calculated using Djikstra's alogrithm [34]. This is true for all of the following algorithms that

consider shortest path.

Next is betweenness centrality. In a graph, there exists a shortest path between any node s and any

other node t. Betweenness centrality of a node i, $C_B(i)$, is the percentage of all shortest paths in

graph $\mathcal{G}$ that traverse node i [20]. It is calculated as:

$$C_B(i) = \sum_{s,t \in \mathcal{V}} \frac{\sigma(s,t|i)}{\sigma(s,t)} \tag{3.7}$$

where $\mathcal{V}$ is the set of all nodes in $\mathcal{G}$, $\sigma(s,t)$ is the number of shortest paths between s and t, and

$\sigma(s,t|i)$ is the number of those paths that pass through i. It is further defined that if $s = t$, then

$\sigma(s,t) = 1$ and if $i \in s,t$, then $\sigma(s,t|i) = 0$.

Degree centrality of a node i, $C_{d,i}$ is simply the percentage of nodes within the graph that are

connected to node i [35]:

$$C_{d,i} = \frac{k_i}{(n-1)} \tag{3.8}$$

where k_i is the degree of node i and n is the number of nodes in graph $\mathcal{G}$.

Current flow closeness centrality, also known as information centrality, is measured for each node. In general, metrics related to current flow centrality differs from the previous centrality measures in that they consider all paths between nodes instead of exclusively shortest paths. Current flow closeness centrality in particular was modeled after how current flows in electrical networks [36]. In circuits, current is distributed over the possible paths; in this metric a similar approach is taken to determine the information content of each path between two nodes. For a graph $\mathcal{G}$ where all nodes are reachable, it is possible to construct a matrix $\boldsymbol{B}$ such that:

$$b_{ii} = 1 + \text{sum of weights of all edges connected to node } i \tag{3.9}$$

$$b_{ij} = 1 - w_{ij} \tag{3.10}$$

where w_{ij} is the weight of the edge connecting nodes i and j. If i and j are not neighbors, w_{ij} is defined to be 0. Denote $\boldsymbol{B}^{-1} = \boldsymbol{C}$. From this matrix, the current flow closeness centrality of node i, I_i, is calculated as:

$$I_i = \frac{n}{nc_{ii} + T - 2R} \tag{3.11}$$

where n is the number of nodes in $\mathcal{G}$, T is the sum of the diagonal elements, $T = \sum_{j=1}^{n} c_{jj}$, and R is the sum of any row in $\boldsymbol{C}$, $R = \sum_{j=1}^{n} c_{ij}$. Note that because the matrix is symmetric, all rows of $\boldsymbol{C}$ will have the same sum, therefore R is not dependent on i [37].

Current flow betweenness centrality, as indicated by the name, also considers all possible paths between the source and target nodes. This measure is also known as random walk betweenness centrality. That is because it represents the expected number of times a random walk will cross node i when it begins at node s and ends at node t. This is calculated as:

$$b_i = \sum_{i \neq s \neq t} r_{st} \tag{3.12}$$

where r_{st} is the element of matrix $\boldsymbol{R}$ which represents the probability that a random walk from s to t passes through i.

Another feature used by the algorithm is communicability centrality, or subgraph centrality [38]. To calculate this measure, consider all of the closed walks in graph $\mathcal{G}$, of length k beginning on node i, denoted by μ_k. A closed walk is any walk that begins and ends on the same node. Repetition of either the edges and vertices is allowed. The communicability centrality of node i, $SC(i)$ is computed as:

$$SC(i) = \sum_{k=1}^{\infty} \frac{\mu_k(i)}{k!} \tag{3.13}$$

The final social network-based features is communicability betweenness centrality. This metric combines the concepts of current flow betweenness centrality and communicability centrality [39]. To compute the communicability betweenness centrality, consider the graph $\mathcal{G}$ with adjacency matrix $\boldsymbol{A}$. Now consider $\mathcal{G}(r)$ which is identical to $\mathcal{G}$ except with all edges connected to node r removed. The adjacency matrix of $\mathcal{G}(r)$ can be written as $\boldsymbol{A} + \boldsymbol{E}(r)$. $\boldsymbol{E}(r)$ has zeros in all elements except for those in the row and column corresponding to node r. The values in these elements are the negative weights of the removed edges, i.e. $\boldsymbol{E}(r)_{ri} = -A_{ri}$. The communicability centrality

for node r is therefore:

$$\omega_r \propto \sum_p \sum_q \frac{\left(e^{\boldsymbol{A}}\right)_{pq} - \left(e^{\boldsymbol{A}+\boldsymbol{E}(r)}\right)_{pq}}{\left(e^{\boldsymbol{A}}\right)_{pq}}, p \neq q \neq r \qquad (3.14)$$

The normalizing constant is omitted for clarity. Note that all metrics used in the learning algorithm were normalized to be between 0 and 1.

All of these different graph statistics were used as inputs into the random forest algorithm to characterize each node's importance in the social graph.

Most Useful Social Network Features

The feature that was ranked the most useful social-network based feature by the feature ranker in Section 4 was the full graph hubs. Recall that a node's hubs value is an estimate of how well a node is connected to authorities. These scores are calculated in an iterative reinforcement algorithm. In the original website ranking context, a hub is a site that links to many sites that are determined to be relevant or important. This can be interpreted to mean that employees with a high hubs score more often communicate with many of the directors or other important employees within the center. The histogram of hub values in the full graph broken down by class is shown in Figure 3.5. This figure shows more class division than Figure 3.2, as graduate students are limited to only the first two bins. It appears that the same employees that had high unique subjects received counts also have large full graph hubs counts, specifically most of the Directors and PMs.

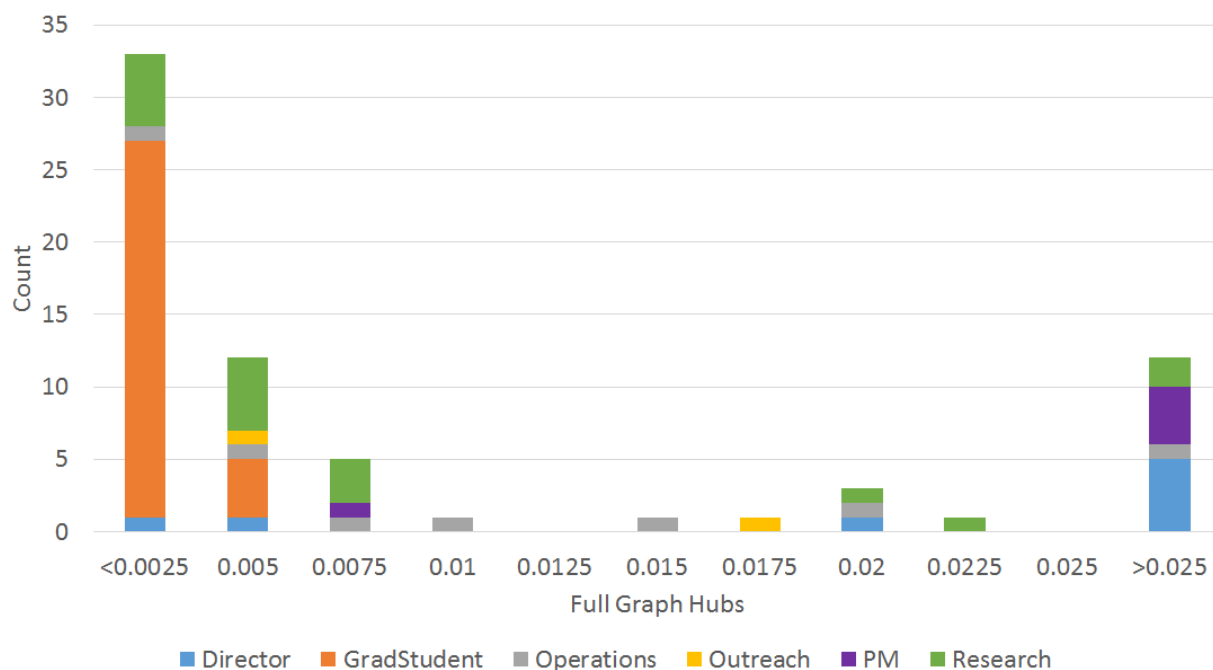The next two most useful measurements were the partial graph communicability centrality and

Figure 3.5: Histogram of hubs from social graph by job title. Note that directors on average have the highest hub score and graduate students have the lowest. In fact, all graduate students have a full graph hubs score < 0.005. All but two PMs have a full graph hubs value > 0.025.

the partial graph communicability betweenness centrality. These measures may seem to have less intuitive interpretations than the top-ranked traffic-based features. However, in general these centrality measures attempt to capture different aspects of how well a node is embedded within the graph. In terms of this application, it measures how the breadth and depth of an employee's email communications within the center.

Chapter 4

Algorithm Design

This chapter describes the analysis behind the main algorithm used in the analysis: the random forest. The first section describes the process of training and cross-validating the model. The second section discusses the method behind ranking the different features used.

4.1   Model Selection

Due to the large number of features and relatively low number of participants, a classification method was carefully chosen to avoid overfitting the data. While tree based classifiers can be susceptible to overfitting, the random forest is robust to overfitting issues and was therefore chosen for this study. The java-based software package Weka was used to generate the random forest
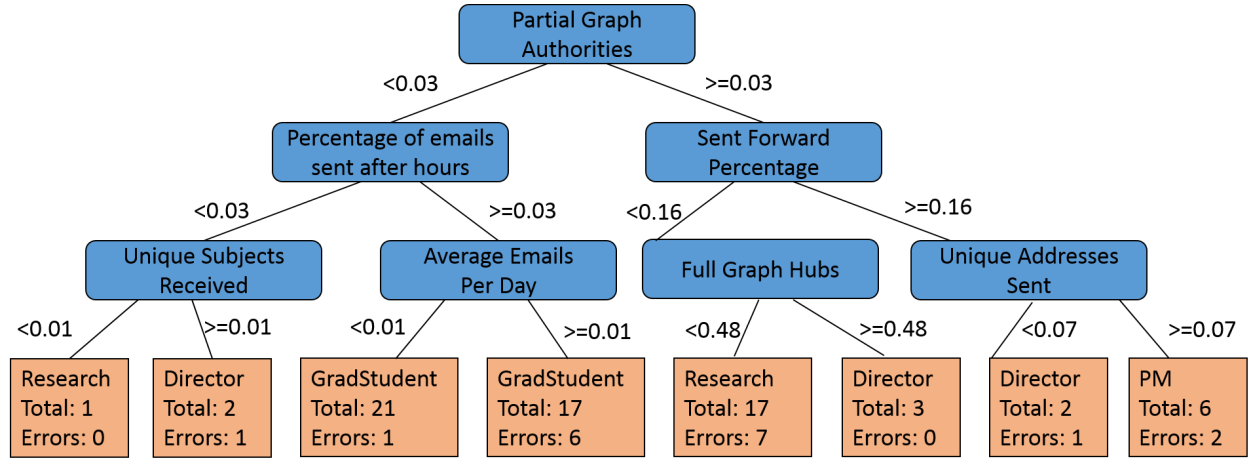
Figure 4.1: Example random tree of depth 3 to demonstrate how a few rules can be used to find significant class divisions.

based on the algorithm described in [40].

Random forests are an ensemble method of machine learning comprised of many random trees. A random tree is a machine learning algorithm that uses training data to learn a series of rules for classification. These rules are constructed in a hierarchy that visually resembles a tree. Each decision is based on what rule will maximize the information gained. An example random tree with depth three is shown in Figure 4.1.

The emails in the dataset were randomly assigned to one of three groups: 35% for training, 30% for cross-validation, and 35% for testing. The training set was used to train the dataset by building the random trees. The cross-validation set was used to select the hyperparameters of the algorithm such as number of trees and number of features used for branch splits. The testing set was used only once on a model trained by the training set using these optimal parameters.

Random forests build many deep random trees with slight random variations. Individually, these random trees overfit the data. However, these random trees are combined through a process of bootstrap aggregating, or bagging. From the cross-validation analysis, the number of trees was 750. The bagging process involves each random tree generating a new training dataset by sampling observations from the input training set with replacement. These subsamples are used to build the random trees. For this analysis, each tree selects $\frac{2N}{3}$ samples to train the trees where N is the number of data points in the overall training set. Just as the samples were subsampled, so were the features. Only this subset of features can be used as rules for that tree. In this implementation, each tree used a subsample of 7 random features. This value represents approximately 6% of the total avaliable features, and this parameter was selected through cross-validation.

After all the trees are built, the test data is run through all the random trees in the forest. Each tree outputs a prediction label for each data point, and the majority vote on each sample is the final predicted label. This random forest model reduces the variance and increases the accuracy of the model compared to a single random tree.

4.2   Feature Selection

Random forests can be difficult to interpret because the ensemble method obscures which features are most meaningful. An attribute analysis helps to better understand which features are better label predictors. Since random trees use information gain to dictate splits, information gain was used as

the evaluation criteria for the features. Specifically, each attribute was evaluated by measuring the information gain with respect to the class. Information gain, also known as mutual information, is calculated as follows:

$$I(\text{Class}; \text{Attribute}) = H(\text{Class}) - H(\text{Class}|\text{Attribute}) \qquad (4.1)$$

where $I(\text{Class}; \text{Attribute})$ represents the mutual information between the class and the attribute, $H(\text{Class})$ is the entropy of the class variable, and $H(\text{Class}|\text{Attribute})$ represents the conditional entropy of the Class given the Attribute value.

Mutual information represents how well knowledge of the attribute informs the prediction of the class. In this model, both the attribute and the class are treated as random variables. The entropy of a random variable is a measure of the uncertainty associated with it. After this information gained value was calculated for each feature, they were ranked in order of most important to least. Table 4.1 shows the top twenty features from this analysis and the features' corresponding information gain. Note that in this table, exactly half of the features are traffic-based and the other half are social-based. This highlights the imporance of using both types of features in this analysis because both types of features contain valuable information related to the class of an individual.

Table 4.1: Top 20 features ranked by the information gain.

Feature	Type	Information Gain
Unique subjects received	Traffic	0.728
Total signed emails received	Traffic	0.728
Partial graph hubs	Social	0.589
Number of emails received as forwards	Traffic	0.0.519
Partial graph current flow closeness centrality	Social	0.0.512
Partial graph pagerank	Social	0.512
Percentage of emails sent with unique addresses	Traffic	0.51
Number of emails received as carbon copies	Traffic	0.5
Full graph pagerank	Social	0.492
Average number of emails received per day	Traffic	0.489
Partial graph communicability betweenness centrality	Social	0.486
Partial graph communicability centrality	Social	0.486
Average number of emails per day (both sent and received)	Traffic	0.479
Partial graph number of cliques	Social	0.47
Percentage of emails received as forwards	Traffic	0.451
Number of emails sent as carbon copies	Traffic	0.43
Partial graph degree centrality	Social	0.429
Partial graph closeness centrality	Social	0.429
Partial graph average shortest paths	Social	0.429
Percentage of signed emails received with unique addresses	Traffic	0.418

Chapter 5

Performance Analysis

The results section first shows the algorithm's ability to correctly classify both the study's volunteers and the additional employees identified from the volunteers' emails. The second part of the results section assumes perfect labeling of the employees and analyzes interactions between employees of different job titles. The ultimate goal of this research was to determine what additional information can be gained by analyzing the organic organizational chart when compared with the official organizational chart.

5.1   Classification Results

As described in Section 4, data was split by randomly assigning each email to training or testing sets with equal probability, 0.35. The remaining 30% of the data was used as a cross-validation set. Then, all of the metrics described in Section 3 were calculated for both groups separately. The training data was used as input to the random forest algorithm as described in Section 4, cross-validation was performed, and predictions were generated for the test data. The number of correct and incorrect classifications for each class are shown below in Figure 5.1. Note that only three predictions were wrong: two people in research were misclassified as graduate students and one graduate student was misclassified as research. It is important to note that two out of the three misclassification are for employees who did not provide their emails for the study. Therefore, the classification accuracy for the study participants is 97.3%, correctly classified inferred employees is 93.8%, and the overall accuracy of this method using all features is 95.7% The confusion matrix for this analysis is another way to visualize the results, as shown in Figure 5.2.

On further inspection, the errors made by the learning algorithm can be explained by understanding center operations. The distinction between graduate students and research faculty is likely the lest clear for two primary reasons. First, some PhD students have been at the center as long or longer than some research faculty. This leads to older PhD students having larger email counts than some of the newer research staff, which could trick the algorithm. Some research faculty are also students, which may cause their email communication patterns to more closely resemble graduate students. To further complicate matters, some graduate students joined the center as staff after they
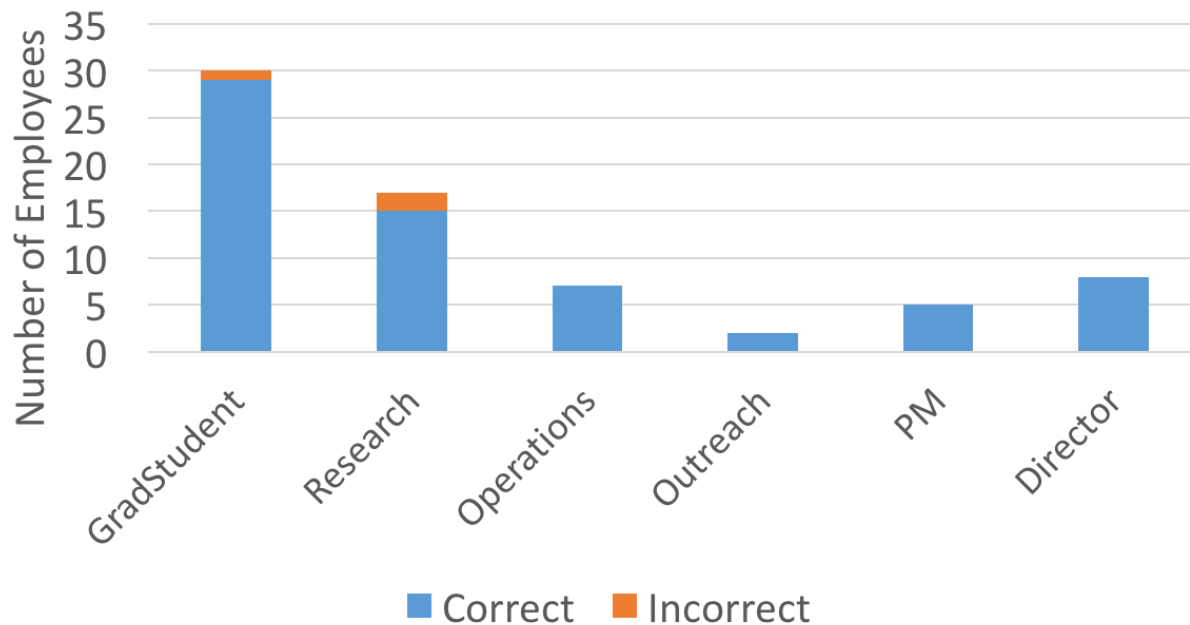
Figure 5.1: The Random Forest algorithm was extremely accurate even for very uneven class sizes. Note that all members of 4 classes were labeled perfectly. There were only 2 errors out of 69 employees, both of which for employees who did not provide emails for the study.

graduated.

Note that this method relies on some assumptions. One is that employees with the same title exhibit similar email behavior. Overall, based on the success of the algorithm and the distributions of the histograms, this seems to prove true. Another premise underlying this analysis is that peoples' email behaviors are consistent over time.

An experiment was performed to evaluate this second assumption. All employees with less than ten months of consistent email behavior were removed from this experiment. Consistent email behavior was defined as sending or receiving at least one email per month. One month of email data

Figure 5.2: Confusion matrix for the job prediction test. The only two classes that confused the classifier were graduate student and research, making a total of 3 errors. These classes are very similar, so it is understandable that there would be errors.

per person was set aside to act as a training set. The random forest algorithm with the parameters described in Section 4 was trained using the first month's data. Each of the nine subsequent months became a separate test set. The features were calculated for each of these months, and the accuracy of the model on the test sets was calculated. The result is a test that evaluates how the average email behavior of the center changes over time, as depicted in Figure 5.3. As the figure shows, there is a decrease in algorithm performance as the time between the training data and the test data increases, but this decrease occurs at a slow rate. Note that with six months separating the training
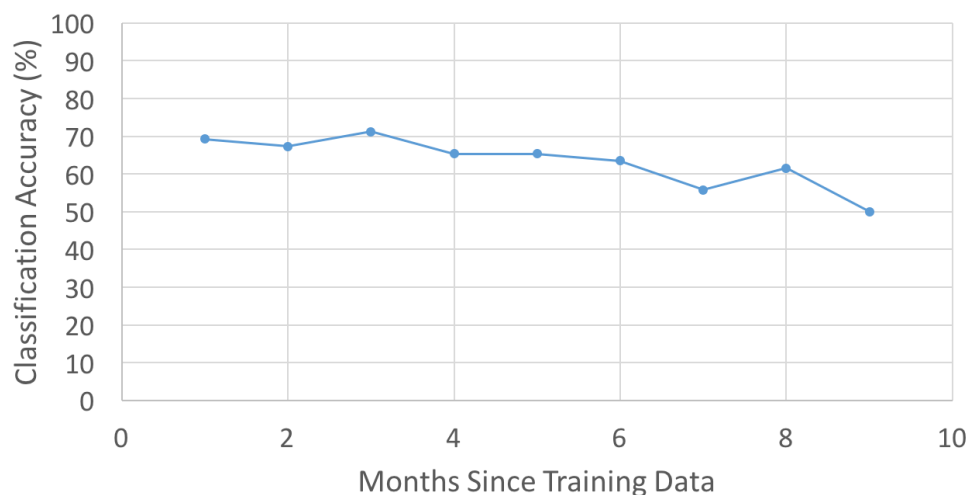
Figure 5.3: As time between the training data and the testing data increases, the accuracy of the prediciton algorithm slowly decreases. This reinforces the assumption that the email behavior of the center employees is relatively constant in time.

data from the test data, the algorithm still performed with 63.4% accuracy. This is a decrease of less than 6% from when the test data is from the month immediately following the training data.

To determine which features were necessary to the analysis, the algorithm was run several times with a subset of the features. The first subset used only the top twenty features from Table 4.1. For each subsequent run, the least useful feature according to the feature analysis was removed from the input to the system until only one feature remained. A plot of this analysis is shown below in Figure 5.4. The maximum accuracy using twenty or fewer features was achieved when either the top fifteen or sixteen were used. For this scenario, there were only 6 errors, resulting in 89.86% accuracy. This is only three more errors than was found using all 114 features. Even using just the top two features resulted in classification accuracy over 86%. Therefore, a very good classifier can be built using much fewer features if the features are selected properly.
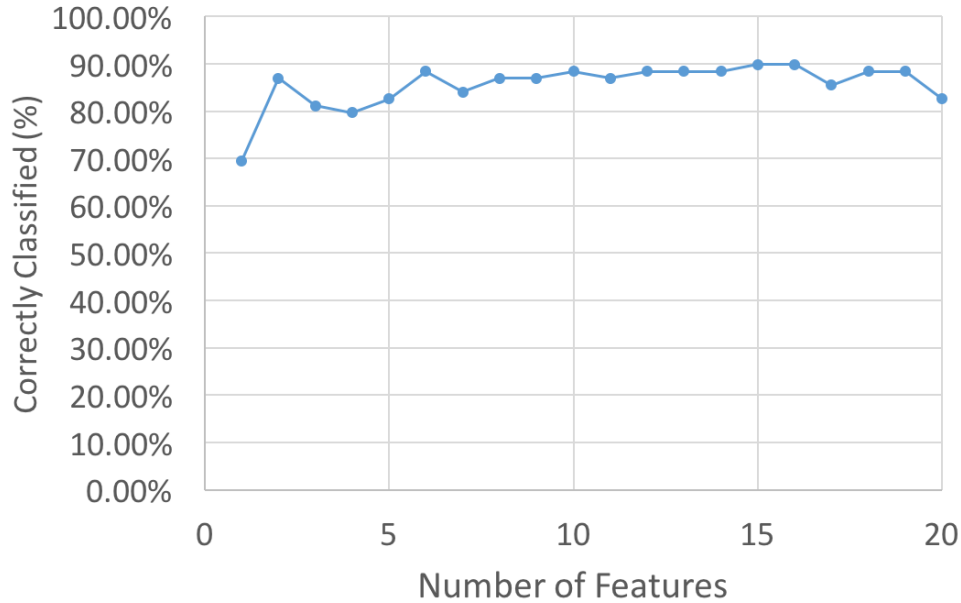
Figure 5.4: Prediction accuracy compared to number of features used for analysis. Note that the accuracy is still very high, 89.86%, when only fifteen features are used. The outcome of using only the top twenty features produces three classification errors, only one more than using the full set of 114 features.

5.2 Leave-One-Out Cross-Validation

Leave-one-out cross-validation (LOOCV) was used as an alternative evaluation method because it removes the time-based assumption. For LOOCV, all of the data is used to calculate the features. Then, all but one person is used as training and the left-out person is used as a single test point. For this experiment, the 2 people of the outreach class were removed, because that would result in only one training point for that class. By testing each point this way, it is possible to evaluate the classification algorithm without having data from all employees in both the training and test sets. This ensures that the algorithm is not learning each person's behaviors individually. The results of this test are reported below in Figure 5.5.

Figure 5.5: Prediction accuracy was higher for graduate students, who have more uniform behaviors and more training data. However, for each category, the majority of the people classified were true members of that group, save for research. Note that even for those classified as research, there were more true researchers than any other class.

This analysis resulted in a lower accuracy than the previous experiment. In total, only 67.2% were correct. The category of graduate student was the easiest to predict. However, note that for each category the majority of the people classified were true members of each group, except for the research category. Even out of those classified as research, there were more true researchers than any other class.

In [12], Namata et al. performed the same analysis on the Enron email corpus. Compared to the dataset used in this analysis the Enron data involved many more people. The Enron dataset has employees distributed more evenly into six classes, one more than used in the analysis above. Even

with these advantages, [12] was only able to achieve 62.09% LOOCV accuracy. The difference between that analysis and the work described in this thesis is that Namata et al. only used traffic-based features. This analysis shows that integrating social-based features with the trafic-based statistics can improve job classification results.

5.3   Hierarchy Analysis

Most of the employees at the center are organized under a director and work with a program manager (unless for example they are a director or program manager). To generate a metric of how well emails can be used to predict the center's organizational chart, the director and project manager for each applicable employee is predicted from the email metadata.

The director of each employee is predicted by the algorithm to be the director that the employee communicated with most by email. Only 57.58% of the center's employees communicate most frequently with their official director. This result points to a possible disconnect between the official organization chart and the organic relationships within the center.

To identify each employee's project manager ground truth is selected to be the project that primarily funds the employee. This time, 72.73% of graduate students and researchers communicate most frequently with their primary program manager. The relation between employees to project managers appears to be stronger than that with directors. Many of the errors in this classification

are due to employees who work with multiple project managers.

Chapter 6

Future Work

There are obvious limitations with this dataset due to its small size. It is also not clear how well these methods and results would generalize to other email datasets, however such datasets are not publicly available. This chapter details what analyses could be possible given a larger dataset. The first section of this chapter presents some analysis showing that more data would improve the classification performance. The second section describes a deep learning method that could be used as a classifier if a massive dataset were available.

6.1   Improving LOOCV Accuracy

As discussed above, the accuracy of the leave-one-out cross-validation test was low because of the limited number of data points. The confusion matrix in Figure 5.5 shows that about one third of the predictions were incorrect. These results can typically be improved by increasing the amount of training data.

An experiment was performed to gauge how the amount of training data affects the accuracy of the LOOCV test. A series of eleven time periods were set for each person in the study, ranging from one week to ten months. Only the 37 employees who sent or received at least one email per week for ten months were included in this analysis, which is a much smaller population to test than what was used in the LOOCV analysis in Ch 5. Some of these 37 employees were direct participants in the study, and some were second-ring employees who did not directly participate in the study. For each employee's time slices, the features were calculated separately and then LOOCV was performed on each time slice. For example, the first run considered the most recent week of email history for each person. The LOOCV accuracy from only one week of data is obviously very low - approximately 32%. Then the test was run on the next time slice, which was the most recent two weeks. This repeated until LOOCV was performed on 10 months of data for each person. The results of this analysis is shown below in Figure 6.1. As the number of days of data increases, the accuracy increases slowly and the variance in the accuracy decreases. A logarithmic curve fit to the data is shown. This shows that more data, as measured by time, in general increases classification accuracy.
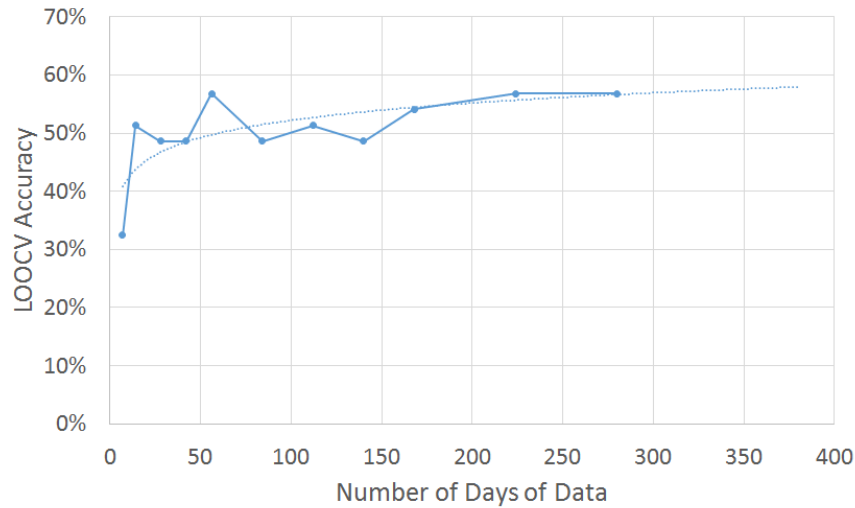
Figure 6.1: Effects of using more data for leave-one-out cross-validation. As the number of days of data increases, the accuracy increases slowly and the variance in the accuracy decreases. This implies that more email data can increase classification accuracy.

The improvement with additional data shown in Figure 6.1 is modest. Recall that the size of the population under test was decreased because of the consistent behavior constraints. In order to investigate the affect of more data, as measured in people, another test was performed. Starting with the original 67 employees (removing the two outreach employees), a random five employees were removed from training data and the LOOCV accuracy was measured for the remaining 62. To combat variance due to the randomness of removing employees, the test was repeated five times and the results averaged together. Then, ten random employees were removed from the original training data and test. This continued until only seven employees remained. These results are shown in Figure 6.2. Similar trends are exhibited between this plot and the previous, namely decreasing variance and increasing accuracy. Again, a logarithmic trendline is shown for reference. It is clear that adding more people, in general, improves the performance of the classifier.
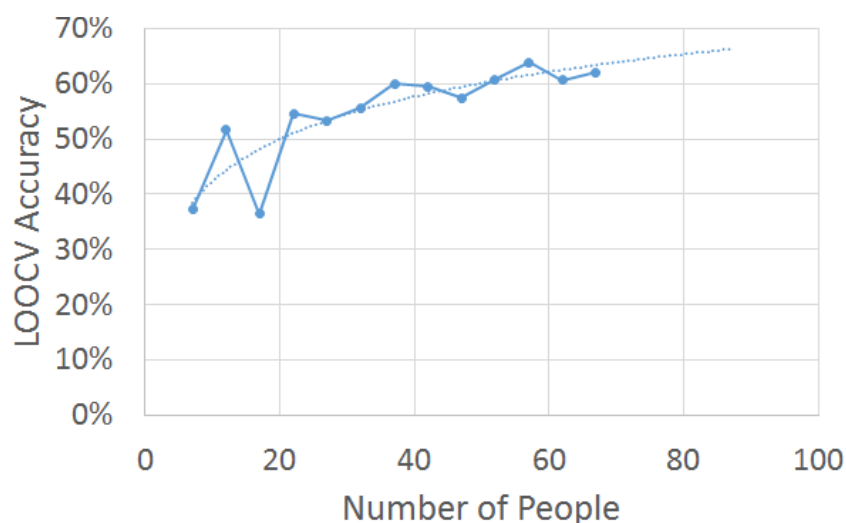
Figure 6.2: Effects of using more people for leave-one-out cross-validation. As more people are included in the training data, the LOOCV accuracy of the classifier increases slowly.

Based on these analyses, it is clear that adding more people to an email dataset and adding people with more historial email data improved the accuracy of the job title classifier.

6.2   Deep Learning Application

Techniques such as random forests work well for small, low-dimensional data. However, if the size of the dataset were greatly increased to billions of people without labels. In this case, a more sophisticated model would be necessary for successful classification in order to accurately learn the complicated structure inherent to such large real-world datasets. Suppose a representative sample of volunteers from the dataset self-identifies themselves. This would produce a set of labeled training data, transforming an unsupervised problem into to a semi-supervised one.
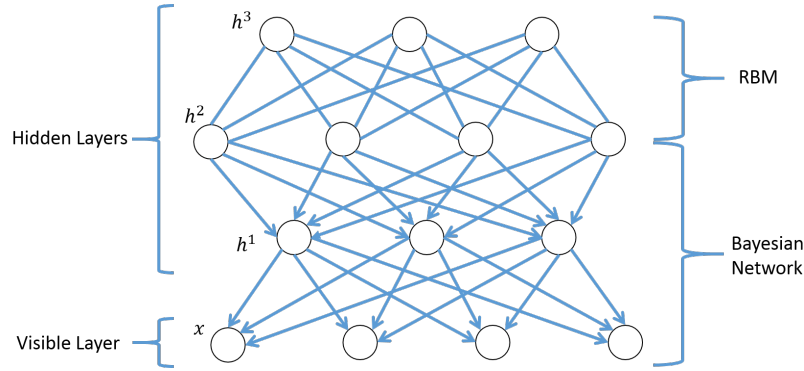
Figure 6.3: A deep belief network is a graphical network constructed by layering Restricted Boltzman machines on top of a Bayesian network [41].

A deep belief network (DBN) is one model used to solve these types of problems. A DBN is a generative graph with nodes that represent stochastic variables. There are several layers of these nodes, some of which are visible and some of which are hidden. Typically, the top two layers of nodes have undirected connections. Therefore, these layers are actually Restricted Boltzman machines (RBMs). The rest of the layers use directed edges, forming a Bayesian network. An example DBN is shown in Figure 6.3.

For the nodes in the Bayesian network layers, there exists a probability of activation, $p(s_i = 1)$, which is represented by the nonlinear sigmoid function applied to a weighted input from the layer above. Specifically,

$$p(s_i = 1) = \frac{1}{1 + \exp\left(-b_i - \sum_j s_j w_{ij}\right)} \tag{6.1}$$

where j represents the ancestors of node i, w_{ij} are the weights on the connections between i and j, and b_i is the bias associated with node i.

The full DBN with l hidden layers is able to model the joint distribution between inputs x and

those hidden layers:

$$P(x, h^1, ..., h^l) = \underbrace{\left(\prod_{k=0}^{l-2} P(h^k|h^{k+1})\right)}_{\text{Bayesian network}} \underbrace{P(h^{l-1}, h^l)}_{\text{RBM}} \tag{6.2}$$

where $P(h^{l-1}, h^l)$ represents the joint distribution of the top two layers, which form the RBM, and

$x = h_0$.

The challenge with very deep networks is that training is very difficult and very time consuming [42]. Unsupervised pre-training can improve the efficency of training deep belief networks, as first described in [43]. In that work, Hinton describes a fast, greedy method to train these networks one layer at a time. This process is as follows:

1. Train an RBM on the input layer, x, and the first hidden layer, h^1.

2. Use the RBM to transform x by sampling $p(h^1|h^0)$ or by finding the mean activations, $p(h^1 = 1|h^0)$.

3. Repeat these steps up through the network until the top layer is reached.

4. Fine-tune all parameters of the deep network

Testing on the MNIST dataset of handwritten digits, unsupervised pre-training is shown to improve classification results over only supervised training [44]. Pre-training identifies a set of initial weights for the network that can ultimately lead to better classification results. An advantage of

unsupervised pre-training is that, with small enough layers, it acts as a regularizer by decreasing variance and increasing the bias [43].

In conjunction with greedy pre-training, backpropagation with some labeled data has been shown to improve both the optimization and generalization of a DBN. Hinton et al. showed that unsupervised pre-training followed by backpropagation of a DBN outperformed traditional feed forward neural networks on the MNIST dataset [41].

Much of the research in deep learning today is focused on image classification and analysis, such as the MNIST dataset. However, DBNs have been applied to text analysis as well. In [45], a sophisticated version of a DBN was used to construct a sentiment classifier. The purpose of that research was to label online reviews as positive, negative, or neutral. The difference between this method and other DBN approaches is that this work replaced some of the hidden layers of the network with a feature selection step. Overall, this improves the training efficiency of the algorithm. This method can classify sentiments more accurately and train more quickly than previous semi-supervised algorithms. A classifier such as this could be used to powerfully learn with extremely large email datasets, particularly those without hashed text.

Chapter 7

Conclusions

This work presents a new dataset, approximately the size of Enron, that was collected from volunteers' emails with particular attention to protect participant privacy. The new dataset includes accurate labels executed by researchers with knowledge of the center and its employees. Statistics were calculated from this dataset, and were used with a random forest algorithm to automatically classify the center's employees. Random Forests are shown to be powerful classifiers by predicting employee job titles with very high accuracy, even for employees for whom only secondhand data is available in the dataset. Using only 3 features, employees are successfully classified higher than 80% of the time and are classified over 95% of the time when 20 features are used. The email data was also used to show that emails could be used to predict an employee's primary program manager, but had a worse chance of being able to identify the director associated with the employee

on the official organizational chart. This work has shown that it is possible to generate important organizational information from using carefully processed email metadata without compromising the privacy of employees.

Chapter 8

Bibliography

[1] S. Radicati and Levenstein, *Email statistics report, 2015-2019*. Technical report, 2015.

[2] D. Fisher, A. J. Brush, E. Gleave, and M. A. Smith, "Revisiting Whittaker & Sidner's email overload ten years later," in *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pp. 309–312, ACM, 2006.

[3] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *Machine learning: ECML 2004*, pp. 217–226, Springer, 2004.

[4] E. M. Bahgat, S. Rady, and W. Gad, "An E-mail Filtering Approach Using Classification Techniques," in *AISI2015, November 28-30, 2015, Beni Suef, Egypt*, vol. 407, pp. 321–331, Cham: Springer International Publishing, 2016.

[5] R. Shams and R. E. Mercer, "Classifying spam emails using text and readability features," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pp. 657–666, IEEE, 2013.

[6] B. He, Z. Li, and N. Yang, "A Novel Approach for Email Clustering Based on Semantics," in *WISA, 2014*, pp. 269–272, Sept. 2014.

[7] P. S. Keila and D. B. Skillicorn, "Structure in the Enron email dataset," *Computational & Mathematical Organization Theory*, vol. 11, no. 3, pp. 183–199, 2005.

[8] Z. Sofershtein and S. Cohen, "Predicting email recipients," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pp. 761–764, ACM, 2015.

[9] Q. Hu, S. Bao, J. Xu, W. Zhou, M. Li, and H. Huang, "Towards building effective email recipient recommendation service," in *SOLI 2012*, pp. 398–403, July 2012.

[10] A. Nordbø, "Data visualization for discovery of digital evidence in email," 2014.

[11] K. K. Waterman and P. J. Bruening, "Big Data analytics: risks and responsibilities," *International Data Privacy Law*, vol. 4, pp. 89–95, May 2014.

[12] G. M. Namata, L. Getoor, and C. Diehl, "Inferring formal titles in organizational email archives," in *Proc. of the ICML Workshop on Statistical Network Analysis*, 2006.

[13] J. Shetty and J. Adibi, "The Enron email dataset database schema and brief statistical report," *Information sciences institute technical report, University of Southern California*, vol. 4, 2004.

[14] E. Gilbert, "Phrases that signal workplace hierarchy," in *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work*, pp. 1037–1046, ACM, 2012.

[15] T. Mitra and E. Gilbert, "Analyzing gossip in workplace email," *ACM SIGWEB Newsletter Winter*, vol. 5, 2013.

[16] G. Tang, J. Pei, and W.-S. Luk, "Email mining: tasks, common techniques, and tools," *Knowledge and Information Systems*, vol. 41, pp. 1–31, June 2013.

[17] K. Yelupula and S. Ramaswamy, "Social network analysis for email classification," in *Proceedings of the 46th Annual Southeast Regional Conference on XX*, pp. 469–474, ACM, 2008.

[18] S. Martin, B. Nelson, A. Sewani, K. Chen, and A. D. Joseph, "Analyzing Behavioral Features for Email Classification.," in *CEAS*, 2005.

[19] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, Nov. 1994.

[20] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[21] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman, "Email as Spectroscopy: Automated Discovery of Community Structure within Organizations," *arXiv:cond-mat/0303264*, Mar. 2003. arXiv: cond-mat/0303264.

[22] M. E. Newman, "A measure of betweenness centrality based on random walks," *Social networks*, vol. 27, no. 1, pp. 39–54, 2005.

[23] A. Agarwal, A. Omuya, A. Harnly, and O. Rambow, "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*, pp. 161–165, Association for Computational Linguistics, 2012.

[24] G. Wilson and W. Banzhaf, "Discovery of email communication networks from the Enron corpus with a genetic algorithm using social network analysis," in *CEC'09*, pp. 3256–3263, IEEE, 2009.

[25] R. Rowe, G. Creamer, S. Hershkop, and S. J. Stolfo, "Automated social hierarchy detection through email network analysis," in *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pp. 109–117, ACM, 2007.

[26] J. Shetty and J. Adibi, "Ex employee status report.," 2004. [http://www.isi.edu/~adibi/Enron/Enron_Employee_Status.xls]. InternetArchive [https://web.archive.org/web/20131126121206/http://www.isi.edu/~adibi/Enron/Enron_Employee_Status.xls], Accessed 1/30/2016.

[27] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, (Pasadena, CA USA), pp. 11–15, Aug. 2008.

[28] J. M. Kleinberg, "Hubs, authorities, and communities," *ACM Computing Surveys (CSUR)*, vol. 31, no. 4es, p. 5, 1999.

[29] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: bringing order to the web.," 1999.

[30] J. Zhang, M. S. Ackerman, and L. Adamic, "Expertise networks in online communities: structure and algorithms," in *Proceedings of the 16th international conference on World Wide Web*, pp. 221–230, ACM, 2007.

[31] J. Saramki, M. Kivel, J.-P. Onnela, K. Kaski, and J. Kertsz, "Generalizations of the clustering coefficient to weighted complex networks," *Physical Review E*, vol. 75, Feb. 2007.

[32] P. G. Lind, M. C. Gonzlez, and H. J. Herrmann, "Cycles and clustering in bipartite networks," *Physical Review E*, vol. 72, Nov. 2005.

[33] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[34] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[35] S. P. Borgatti and D. S. Halgin, "Analyzing affiliation networks," *The Sage handbook of social network analysis*, pp. 417–433, 2011.

[36] U. Brandes and D. Fleischer, *Centrality measures based on current flow*. Springer, 2005.

[37] K. Stephenson and M. Zelen, "Rethinking centrality: Methods and examples," *Social Networks*, vol. 11, no. 1, pp. 1–37, 1989.

[38] E. Estrada and J. A. Rodríguez-Velázquez, "Subgraph centrality in asdf," *Phys. Rev. E*, vol. 71, p. 056103, May 2005.

[39] E. Estrada and N. Hatano, "Communicability in complex networks," *Physical Review E*, vol. 77, no. 3, p. 036111, 2008.

[40] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[41] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[42] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[43] G. E. Hinton, "To recognize shapes, first learn to generate images," *Progress in brain research*, vol. 165, pp. 535–547, 2007.

[44] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?," *The Journal of Machine Learning Research*, vol. 11, pp. 625–660, 2010.

[45] P. Ruangkanokmas, T. Achalakul, and K. Akkarajitsakul, "Deep belief networks with feature selection for sentiment classification,"