Hierarchy detection through email network analysis

Michal Duczynski
University of Warsaw
m.r.duczynski@student.vu.nl

Bojian Yin
Vrije University Amsterdam
byn800@vu.nl

ABSTRACT

Email is an indispensable network communication tool both in our daily life and working.The role information of a specified community entity can be discovered by data mining. This paper we will build the hierarchy structure of ENRON Corporation by ENRON email corpus. To resolve this problem, we firstly resolve role recognition problem of Email network, through Social Network Analysis tools and based on analysis of communication behaviors and email content. Then, use functional hierarchy building algorithm and Greedy algorithm to structure the hierarchy tree and apply evaluation functions to estimate the accuracy of the hierarchy relationships . Enron's top managers' organization structure is set up based on SNA and email dataset. The analysis result have a certain reference value and practical significance.

Keywords

Social Network Analysis, Enron Email, Hierarchy Structure

1. INTRODUCTION

Since the origin of the social network, it had a re-definition and cultures are varied, but, the key technological features – (1) 'keep in touch with others' and (2) 'identity management' – are consistent. And Social Network sweep the world in a short period. More and more of life is now displayed online, and more and more digital traces are generated by online activities.

Social network is a combination of a number of individuals, or organizations and the connection between them. Through the study of social network theory, we can try to discover the relationships behind these communications, and apply it to information recommendation, e-commerce and so on. Emails, as an important part of the social network within companies, has become the most efficient communication platform for cooperation and informations sharing. However, how to figure the hierarchy structure as well as core person such as CEO form a great deal of dataset is a tough problem.

To investigate the social structure, social network analysis combines both network and graph theories and has become one of the key techniques in sociology and management sciences. The network is composed of nodes representing individuals and edges representing interactions and relations between individuals.

As we will try quantify the communication between as was already done by several other analyzes...

Enron was a famous energy company in the last decades, but it went bankrupt within several weeks in 2002. Financial fraud, more specifically insider trading, rarely cause a company to go bankrupt, however the financial fraud at Enron was well-designed and institutionalized having taken place for several years. What's more the direct involvement of higher management made attracted the focus of the public and led to the scandal.

Research on hierarchy structure was presented on this paper based on the email communication history

1.1 Dataset description

The Enron Corporation's email dataset is an online source which is publicly available set. And this data was collected and prepared by the CALO project during the judicial processing against the Enron Corporation in 2002. Then, to reuse the dataset on social researching, email dataset was purchased by MIT. Several research on NLP (natural language processing) and SNA is based on this corpus.

2. RELATED WORK

Let us start the related work section with: in the future work directions Hardin et al. (2014) said "We are not aware of any studies that carefully interpret the significance of high rank in centrality measures in the context of the company's hierarchy"[4].

Zhou et al. proposed an algorithm to create a hierarchy structure based on a local measure of number of emails exchanged between two employees rather than the whole graph [7]. They haven't performed a quantitative evaluation of their algorithm. As one of the experiments we carried out, we've implemented their algorithm and evaluated it, a more detailed description can be found in the Experiments section.

Creamer et al. proposed a score S, which represents *social* hierarchy importance [2]. The $score \in [0, 1]$, is defined as a weighted average of a variety of metrics from SNA, among others: network centralities, clustering coefficient, number of

cliques an employee belongs to, average response time. They haven't published the weights for their *score* saying that the appropriate contribution, i.e. weight, of each (measure) will vary by situation and organization, and therefore can be adjusted to achieve more accurate results in a variety of cases. Once they calculated the *score* for each employee they, partitioned the employees into levels by putting all the employees with score from 0.8-1.0 on the top level, users with social scores from 0.6-0.8 are placed on the next level down and so on. To evaluate the quality of such partition they compared the distribution of job titles on each level, they reported that indeed the average level of senior manager titles (CEO, director) is higher than the average level for the regular employees.

Agarwal has manually compiled a *gold standard* for Enron organization hierarchy [1]. Thanks to the *gold* hierarchy they were able to compare two approaches to a problem of predicting the direction of dominance relation between a pair of employees. Obviously the baseline solution is to guess the dominance relation, this will yield an accuracy of 0.5. As their approach they have decided to pick a simple measure of degree centrality, so for a pair of employees connected by a dominance relation, the supervisor is the one with higher degree centrality, this approach resulted in an accuracy of 87.58%. In comparison a sophisticated NLP approach that identified phrases which signal workplace hierarchy [3] achieved an accuracy of 82.37%.

3. RESEARCH QUESTIONS

- Analyzing Enron email corpus doesn't require *Big Data* tools.
- The hierarchy of Enron can be reliable reconstructed from the employees' emails.
- The hierarchy should be reconstructed by looking at emails exchanged not only between the core employees.
- If a person A is connected to two people B, C with comparably high importance scores, the A's supervisor is the person to whom A is more strongly connected.
- Employees high in the reconstructed hierarchy were important to the scandal.

4. SOCIAL NETWORK ANALYSIS

A social structure made of nodes or individuals that are related to each other by various interdepedencies like friendships, administration, etc.

4.1 Social capital

On the employee level, social capital could define as the potential resource that an individual controlled within a social network. The social capital is a different conception compare with economical capital or other capitals. The difference is that the social capital is behind in the social relationships. Individuals cannot govern the social capital like what they do for other capitals. The cooperation with other individuals is the essential for the use of social capital.

Structural hole is the gap within the social network and individuals around the hole cannot directly connect with others. However, powerful or influential people could control the information stream if he or she bridged entities around the hole.

Individuals could increase their social capital by linking or bridging the un-connected groups.

4.2 Overview of centrality measures

We would like to introduce the three centrality measures which we will be using throughout this paper and on which we've built our algorithm.

For the sake of completeness we have included tables for each centrality showing the top ten employees according to a given centrality in the Appendix.

4.2.1 Degree Centrality

Degree centrality is the first one and the easiest one. Degree is the most direct way to measure the importance of a node in the network, as it is defined as the number of links the node has:

$$C_D(v) = \deg(v) \tag{1}$$

4.2.2 Closeness Centrality

Closeness centrality of a node u is the reciprocal of the sum of the shortest path distances from u to all $n-1$ other nodes. As the sum of distances depends on the number of nodes in the graph, closeness can be normalized by looking at the mean distance instead of the sum. So the formula to calculate closeness centrality is as follows:

- Average Distance:

$$D_{avg}(v) = \frac{1}{|V|-1} \sum_{u \neq v \in V} d(v,u) \tag{2}$$

- Closeness Centrality:

$$C_{Closeness}(v) = (D_{avg}(v))^{-1} = \frac{|V|-1}{\sum_{u \neq v \in V} d(v,u)} \tag{3}$$

Intuitively we can see high closeness of a node equates to it being in the center of the graph, however it doesn't say anything about how crucial the node is, it might happen that the removal of the node doesn't affect the speed of information diffusion.

4.2.3 Betweenness Centrality

A node's betweenness is the number of shortest paths between other nodes that pass one node. The formula to calculate the betweenness centrality $C_B$is:

$$C_B(v) = \sum_{s \neq v \neq t \in V, s<t} \frac{\zeta_{st}(v)}{\zeta_{st}}$$

where ζ_{st} is the number of shortest paths between node s and t, and $\zeta_{st}(v)$ is the number of shortest paths between node s and t that pass through v.

While describing closeness centrality we mentioned that the removal of a node with high closeness might not affect the speed of information diffusion, this is in contrast to the betweenness. Removing a node with betweenness higher than 0 will affect the speed of information diffusion.

5. PROJECT SETUP

5.1 Choice of Enron dataset

Since 2004 when the original Enron email dataset was published, the official version has been not only modified to remove sensitive messages not related to the scandal, but also researchers have created a cleaned up version of the corpus available for example as MongoDB databases[1]. As a practice exercise we have decided to use the official version without attachments [1], which is 1.42GB of plain text emails, spread through folders for each core employee. We haven't used the version available on the cluster (50GB compressed), as we didn't consider attachments to be important for our analysis.

5.2 Cleaning the data

The data was cleaned and analyzed in Python. Following steps were taken to clean-up Enron data:

- parsing emails using Python's `email` module.
- identifying email aliases for each core person
- removing duplicates by keeping an md5 hash of `sender#recipient#content`
- serializing only the email metadata: FROM, TO, CC and Date fields, in JSON for further processing.

Key statistics:

- 491 thousand emails
- 234 thousand unique emails
- 52% percent of emails are duplicate
- graph, core, 4 244 edges
- graph, all, 6 388 248 edges
- around 10 minutes to cleanup the data on a commodity laptop

6. EXPERIMENTS

6.1 Problem statement

Recreate organisational hierarchy tree based on email corpus.

6.2 Email network graph

We will create an email network graph $G = (V, E, w)$, by looking at the email metadata, more specifically at the fields: From, To, CC list.

6.2.1 Set of nodes

There are two natural choices for the set of nodes:

1. only the nodes for *core* 156 Enron employees, as we are predicting the hierarchy only for them
2. a node for every employee

Obviously the set of nodes we choose, significantly alters the graph and centrality measures. The biggest difference can be observed for degree centrality, if the graph is limited to only the *core* employees the degree centrality could theoretically reach 156, this doesn't hold anymore for the graph of all employees.

[1]https://www.cs.cmu.edu/ ./enron/

6.2.2 Edge weight

There is a variety of ways for defining the edge weight.

- Hardin et al. [4] defined the edge weight w as:

$$w(u,v) = |M_{u,v}| + \sum_{c \in C_{u,v}} \frac{1}{\sqrt{1+|c|}} \tag{4}$$

  where $M_{u,v}$ is the set of emails sent from u to v and $C_{u,v}$ is a multi-set of CC lists that contain both u and v, as a result the fewer people are mentioned by an email the greater this email contribution to the score of mentioned people.

- De Choudhury et al. [6] took the geometric mean of sent-received counts as the weight, $w(u,v) = \sqrt{M_{u,v} \cdot M_{v,u}}$.
- Zhou et al. defined the edge weight $w(u,v)$ more simply, as the number of emails that mention u and v in any of the fields [7].
- Agarwal et al. considered an un-weighted graph and added a link between two employees if one sends at least one email to the other (who can be a TO, CC, or BCC recipient) [1].
- We wanted to experiment with edge weight, as given by the cosine similarity:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{5}$$

  where $A_i = 1$ if the user A was mentioned by i-th email, otherwise $A_i = 0$.

6.2.3 Filtering edges

In their research De Choudhury et al. analyzed the effect of filtering out the edges with weight below a threshold τ on graph structure for Enron data [6]. Their conclusions were there isn't a clearly optimal value for the threshold.

We have discovered that a different approach to filtering edges produces better results for visualizing Enron network. We used cosine similarity as the edge weight for the visualization. Without edge filtering the resulting network was unreadable, as everyone seemed connected to everyone and no clear structure was visible. We tested different values for the cut-off threshold τ, but the people highest in the hierarchy had low similarity values with other employees. This means they are the ones to become disconnected first as the τ increases.

That's why we've chosen a different approach where each node marks to keep edges with n most similar neighbors. After this filtering a node u can have more than n neighbors, as each neighbor of u could have marked to keep u as its neighbor.

6.3 Gold standard for Enron organisational hierarchy

Most previous research either haven't performed quantitative evaluation or only looked at job titles, for example, trying to predict the quality of the hierarchy by counting the number of dominance pairs in which the person being supervised has a higher job title than the supervisor [7][2][1].

Figure 1: A visual depiction of the hierarchy of Enron job titles.[3]

To enable more accurate evaluation of Enron hierarchy prediction Apoorv et al. have complied a gold-standard hierarchy, which they extracted manually from pdf files (with names similar to "org-chart.pdf")[1]. The gold-standard contains information for 1,518 employees, this translates to 13,724 dominance pairs. As this dataset is freely available upon request, we have obtained it from Apoorv. The dataset describes 3 types of relations:

1. person A supervises person B
2. team T contains person A
3. person A manages team T

Our goal was to build a hierarchy tree for the core employees, to do that we have joined the relations (supervise, contain and manage) together and filtered out nodes other than core employees. In more detail, a supervise relation: $A \to b \to c \to D$, becomes $A \to D$, where A and D are core employees and b, c aren't. As we have checked using only relation supervise doesn't create one tree but many small trees. This could be explained by that fact we were able to find any relation information for 82 out of 156 core employees. What's more there are some instances when a person belongs to two teams and thus when joining the three relations together it may happen that a person will have two supervisors. As can be seen on figure 4, out of the 58 people 4 people have 2 two supervisors. To make sure the hierarchy extracted from the *gold-standard* makes sense, we've checked it manually.

6.4 Evaluation metric

The evaluation metric will be based on the similarity of predicted hierarchy tree and gold standard hierarchy tree [1]. There is no widely accepted tree similarity metric. Let P be the relation of being a direct supervisor, thus for a directed graph $E(u, v) = P(u, v)$. Let P^* denote the transitive closure of P, thus the dominance relation.

The dominance relation labels each possible dominance pair as positive, when the dominance occurs and negative when it doesn't. This shows we can treat the task of comparing the predicted dominance relation P and true dominance relation P_{True} as a binary classification problem. In this framework the True Positives are $P^* \cap P^*_{True}$, the True Negatives are $\overline{P^*} \cap \overline{P^*_{True}}$. This creates a classic confusion matrix.

There are several measures to express the predictive quality, for instance Accuracy (True Positives + True Negatives / Total Population) and Recall (True Positive / Condition Positive). Depending on which measure we want to optimize we will construct the algorithm differently later on.

Let us analyze Accuracy measure first, because it takes into account both True Positives and True Negatives an algorithm optimized for Accuracy may return a set of trees, since joining the trees may require adding edges which are False Positives, thus decreasing the True Negative rate and Accuracy.

We don't run into the same problem if we use Recall, joining several trees together can't decrease the Recall. It can on increase it as we add more dominance pairs that may happen to be true. This is the reason why we will be using Recall for our evaluation.

$$Recall(P, P_{True}) = \frac{|P^* \cap P^*_{True}|}{|P^*_{True}|} \tag{6}$$

A different statistic for evaluating a hierarchy was proposed by Creamer et al., for each role (e.g. CEO or trader) they calculated the mean level in hierarchy of people with this role [2].

6.5 Recall of a random hierarchy

Because the problem of recreating a hierarchy is quite hard, personally we didn't have any intuition of what score to expect from a random solution. Comparing your solution to the random one is important as it might happen that centrality measures are negatively correlated with position in the true hierarchy and also it quantifies how hard your problem is. In a problem where random solution scores 0.50 accuracy your algorithm should score at least 0.51.

The mean recall of 100 random hierarchy trees is only 0.027.

The random trees were built in a following way, the list of nodes were permuted and sequentially a next node from the permuted list of nodes were attached to any of the nodes already present in the tree.

6.6 Functional hierarchy building algorithm

We will introduce a template for a hierarchy tree building algorithm. The algorithm takes a weighted un-directed graph as input. Let N_u be the set of neighbors of u.

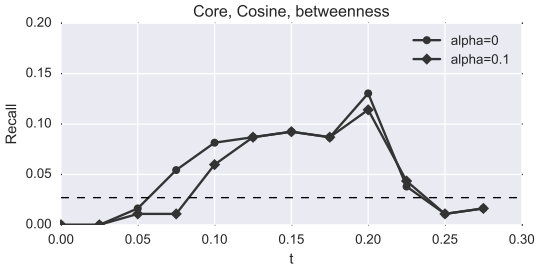
$$g(u) = u \geq max_{v \in N_u} C(v) \tag{7}$$

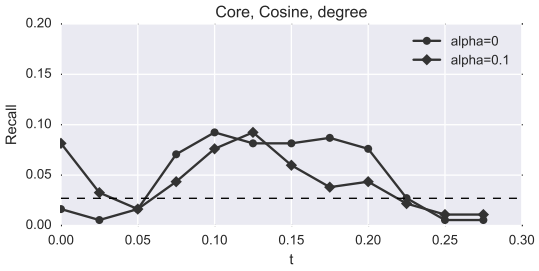
$$Parent(u) = \begin{cases} u & g(u) \\ \arg\max_{v \in N_u} \alpha w(u, v) + (1 - \alpha) C(v) & !g(u) \end{cases} \tag{8}$$

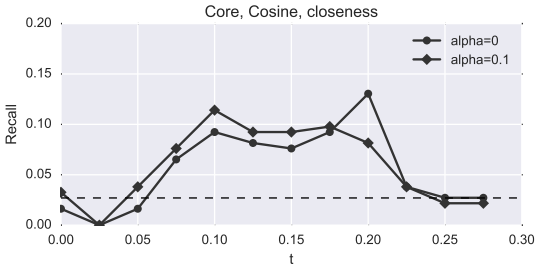
where $w(u, v)$ is the weight of the edge (u, v), $\alpha \in [0, 1]$ is a trade-off factor between similarity and centrality, C is based on a centrality.

The algorithm can return a set of trees in two cases, when the graph is composed of multiple connected components or a node has centrality equal to the maximum centrality of its neighbors. Because our goal is to maximize Recall and we can always connect the trees together without decreasing Recall, the final step of the algorithm is to connect the roots of trees to the root of the biggest tree.

6.6.1 Evaluation







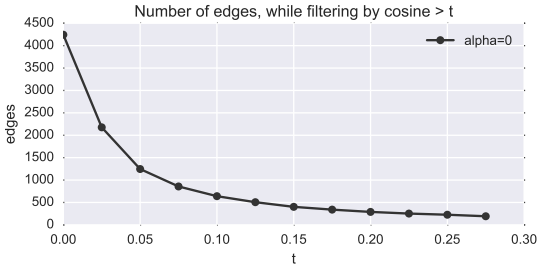
The *true* dominance relation we prepared based on the gold standard contains 184 dominance pairs for 58 employees. Our best result had a recall around 14%, this means our hierarchy had only 26 true dominance pairs. However the recall of a random hierarchy is only 2.7%, so our algorithm performs significantly better than chance. Having said that, the inescapable conclusion is we weren't able to reliably predict organizational hierarchy based on centrality measures.

Our another finding is that while choosing between possible supervisors $S(A)$ for node A the approach of looking at the centrality and taking into account how strongly A is connected to each of them produces comparable or slightly worse results than ignoring the connection strength (aka weight) for degree and betweenness centralities. The exceptions to this are closeness centrality and degree centrality without filtering weak connections (edges).

By looking at the plots it becomes apparent that the there is a certain amount of edge filtering that produces the best results which is consistent for the three centrality measures we tested. Notably as the filtered graph contains from 289 to 640 edges (t=0.2 and 0.1 accordingly, 640 edges translates to average degree of 8.5).

The way filtering edges has significant impact on the algorithm can be easily explained by the fact we are using *traditional* un-weighted centrality measures, and not the weighted centrality measures, which were introduced in 2010 [5].

For the sake of completeness we have also evaluated our algorithm for a graph with nodes for all employees, the recall is consistently lower and only sightly better than chance (baseline of random hierarchy), see figure 2 in Appendix.



6.7 Greedy hierarchy building algorithm

Zhou et al. proposed a greedy algorithm for a weighted directed graph, which adds edges with highest directionality first [7].

```
g = DirectedGraph()
has_parent = set()
while pairs.not_empty():
   (u, v), score = pairs.pop_with_max_score()
   if v not in has_parent:
       g.add_edge(u, v)
       has_parent.add(v)
```

The algorithm can be written equivalently in a functional form. For a weight w describing the directionality of supervision relation between u and v, where $w(u,v) = -w(v,u)$, the algorithm is:

$$Parent(u) = \begin{cases} u & \max_{v \in N_u} w(v,u) < 0 \\ \arg\max_{v \in N_u} w(u,v) & \text{otherwise} \end{cases} \tag{9}$$

If we define $w(u,v) = C(u) - C(v)$, then the algorithm becomes identical to the "functional" hierarchy algorithm with $\alpha = 0$.

6.7.1 Evaluation

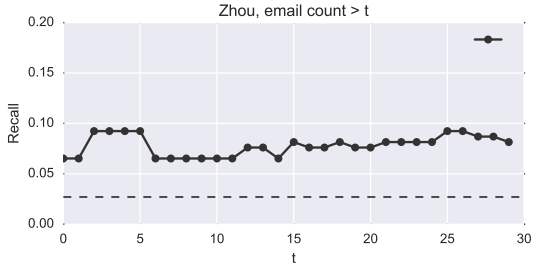
Zhou et al. [7] published his paper in 2005 before the gold standard for ENRON hierarchy [1] was prepared in 2012, thus his work didn't include a quantified evaluation. We have implemented and evaluated his approach.

Zhou et al. [7] considered a heuristic based on following formulae:

$$P(u \mid v) = \frac{support(u \cap v)}{support(v)}$$

$$\begin{aligned} Supervises(u,v) &= P(u \mid v) - P(v \mid u) \\ &= support(u \cap v) \left(\frac{1}{support(v)} - \frac{1}{support(u)} \right) \end{aligned}$$

Intuitively the formula can be interpreted as u is likely to supervise v if they exchanged plenty of emails and u is mentioned by considerably more emails than v. It is worth noting that $Supervises(u,v) = w(u,v)$ for the algorithm above.



One of the obvious observation is that the Zhou's algorithm isn't as sensitive to the amount of edge filtering as our algorithm, this can be easily explained by the fact that Zhou uses a local measure based on the amount of emails exchanged by the two users, as opposed to a centrality which is based on the graph as a whole. More directly an edge with a high score will most likely result in an edge in the hierarchy tree and this edge is unlikely to get filtered once again because of its high score.

Another observation is the fact that our algorithm performs significantly better than Zhou's (recall of around 14% vs 10%).

7. VISUALIZATION

We used D3.js for the two of our visualizations, the hierarchy tree and connection graph. The hierarchy tree is linked to the connection graph, by clicking on a node in the tree the connection graph with edges highlighted for this node will be shown. As already described in the edge filtering section we used cosine similarity for the strength of connection between the users additionally to make the graph more readable we are displaying only the connections with 6 most *similar* employees for each employee. The people facing convictions are marked red on the hierarchy tree.

8. CONCLUSIONS

We have shown that social network analysis of Enron email corpus can be done on a commodity laptop without the use of any Big Data tools.

We have shown the problem of reconstructing the organizational hierarchy is hard by calculating the recall (2.7%) of a baseline solution that is of a random hierarchy. Our algorithm based on centrality measures performs siginifcantly better than chance (recall of 14%) and better than the algorithm introduced by Zhou (recall of 10%). However considering the recall scores are unarguably low, we have to say that the organizational hierarchy of Enron can not be reliably reconstructed from employees' emails.

By comparing the results for algorithm running on the graph with nodes for all employees versus the graph with nodes for only the core employees, it becomes apparent the hierarchy reconstruction should be performed only based on emails exchanged between core employees ignoring other emails to non-core employees.

An indisputable observation is the importance of edge filtering for the performance of an algorithm based on centrality measures. The optimal level of filtering we have found for Enron corpus removes more than 85% of edges (640 edges for t=0.1 to 4244 edges in total means 15% of edges are left intact).

Another way of taking into account edge weight was the idea to break a tie, when multiple our neighbors have identical centrality score by choosing the one with whom we have a stronger connection. Our results demonstrate this has minimal influence on the score and can be omitted from future work.

Finally by looking at the generated hierarchy tree, figure 5, it is hard to observe any correlation between the position in hierarchy and importance to the Enron scandal.

9. REFERENCES

Table 1: All, email count, t=10, degree

Rank	Employee	Role	score
0	beck-s	Employee	0.21
1	kean-s	Vice President	0.20
2	chris.mallory@enron.com	None	0.17
3	allen-p	N/A	0.16
4	lisa.gillette@enron.com	None	0.16
5	lavorato-j	CEO	0.16
6	shively-h	Vice President	0.16
7	tracy.ngo@enron.com	None	0.16
8	leslie.reeves@enron.com	None	0.16
9	christian.yoder@enron.com	None	0.16

[1] A. Agarwal, A. Omuya, A. Harnly, and O. Rambow. A comprehensive gold standard for the enron organizational hierarchy. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 161–165. The Association for Computer Linguistics, 2012.

[2] G. Creamer, R. Rowe, S. Hershkop, and S. J. Stolfo. Segmentation and automated social hierarchy detection through email network analysis. In H. Zhang, M. Spiliopoulou, B. Mobasher, C. L. Giles, A. McCallum, O. Nasraoui, J. Srivastava, and J. Yen, editors, *Advances in Web Mining and Web Usage Analysis, 9th International Workshop on Knowledge Discovery on the Web, WebKDD 2007, and 1st International Workshop on Social Networks Analysis, SNA-KDD 2007, San Jose, CA, USA, August 12-15, 2007. Revised Papers*, volume 5439 of *Lecture Notes in Computer Science*, pages 40–58. Springer, 2007.

[3] E. Gilbert. Phrases that signal workplace hierarchy. In S. E. Poltrock, C. Simone, J. Grudin, G. Mark, and J. Riedl, editors, *CSCW '12 Computer Supported Cooperative Work, Seattle, WA, USA, February 11-15, 2012*, pages 1037–1046. ACM, 2012.

[4] J. Hardin, G. Sarkis, and P. C. Urc. Network analysis with the enron email corpus. *CoRR*, abs/1410.2759, 2014.

[5] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 32(3):245–251, 2010.

[6] G. Tang, J. Pei, and W. Luk. Email mining: tasks, common techniques, and tools. *Knowl. Inf. Syst.*, 41(1):1–31, 2014.

[7] D. Zhou, Y. Song, H. Zha, and Y. Zhang. Towards discovering organizational structure from email corpus. In M. A. Wani, M. G. Milanova, L. A. Kurgan, M. Reformat, and K. Hafeez, editors, *Fourth International Conference on Machine Learning and Applications, ICMLA 2005, Los Angeles, California, USA, 15-17 December 2005*. IEEE Computer Society, 2005.

APPENDIX

Core, email count, t=0, 4244 edges.

Table 2: All, email count, t=10, degree, filtered to show core

Rank	Employee	Role	score
0	beck-s	Employee	0.21
1	kean-s	Vice President	0.20
2	allen-p	N/A	0.16
3	lavorato-j	CEO	0.16
4	shively-h	Vice President	0.16
5	jones-t	N/A	0.15
6	presto-k	Vice President	0.15
7	shankman-j	President	0.14
8	shapiro-r	Vice President	0.14
9	hodge-j	Managing Director	0.14

Table 3: Core, email count, t=0, degree

Rank	Employee	Role	score
0	allen-p	N/A	0.78
1	presto-k	Vice President	0.78
2	lavorato-j	CEO	0.77
3	swerzbin-m	Trader	0.74
4	shively-h	Vice President	0.72
5	grigsby-m	Manager	0.72
6	sturm-f	Vice President	0.72
7	arnold-j	Vice President	0.72
8	arora-h	Vice President	0.71
9	neal-s	Vice President	0.71

Table 4: Core, email count, t=0, closeness

Rank	Employee	Role	score
0	allen-p	N/A	0.82
1	presto-k	Vice President	0.82
2	lavorato-j	CEO	0.82
3	swerzbin-m	Trader	0.80
4	shively-h	Vice President	0.78
5	grigsby-m	Manager	0.78
6	sturm-f	Vice President	0.78
7	arnold-j	Vice President	0.78
8	arora-h	Vice President	0.77
9	neal-s	Vice President	0.77

Table 5: Core, email count, t=0, betweenness

Rank	Employee	Role	score
0	scott-s	N/A	0.02
1	williams-w3	Director	0.02
2	dean-c	Trader	0.02
3	scholtes-d	Trader	0.02
4	zufferli-j	Employee	0.02
5	presto-k	Vice President	0.01
6	swerzbin-m	Trader	0.01
7	forney-j	Manager	0.01
8	campbell-l	N/A	0.01
9	causholli-m	Employee	0.01

Table 6: Core, cosine sim, t=0.1, closeness

Rank	Employee	Role	score
0	allen-p	N/A	0.43
1	shively-h	Vice President	0.42
2	presto-k	Vice President	0.41
3	sturm-f	Vice President	0.41
4	neal-s	Vice President	0.41
5	martin-t	Vice President	0.40
6	swerzbin-m	Trader	0.40
7	arora-h	Vice President	0.40
8	lavorato-j	CEO	0.39
9	davis-d	N/A	0.38

Table 7: Core, cosine sim, t=0.1, degree

Rank	Employee	Role	score
0	quenet-j	Trader	0.19
1	allen-p	N/A	0.19
2	donohoe-t	Employee	0.18
3	neal-s	Vice President	0.18
4	shively-h	Vice President	0.16
5	sturm-f	Vice President	0.16
6	martin-t	Vice President	0.15
7	grigsby-m	Manager	0.14
8	mckay-b	Director	0.14
9	benson-r	Director	0.13

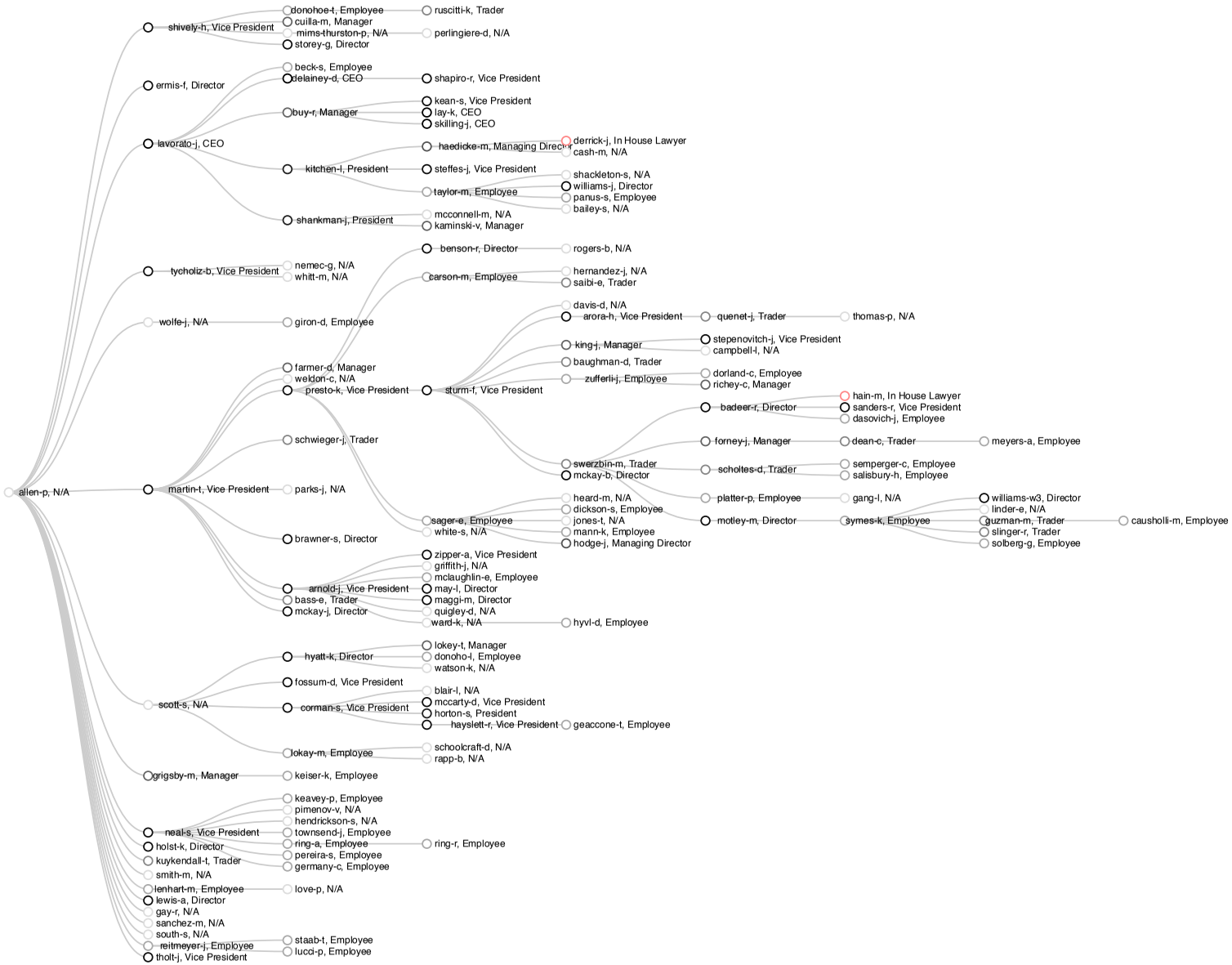


Figure 2: Hierarchy as generated by functional hierarchy building algorithm for closeness centrality with cosine similarity > 0.1, α=0.1

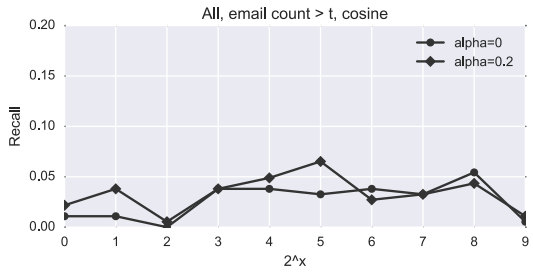


Figure 3: Results for functional hierarchy building algorithm, for all employees (not only core), based on degree centrality

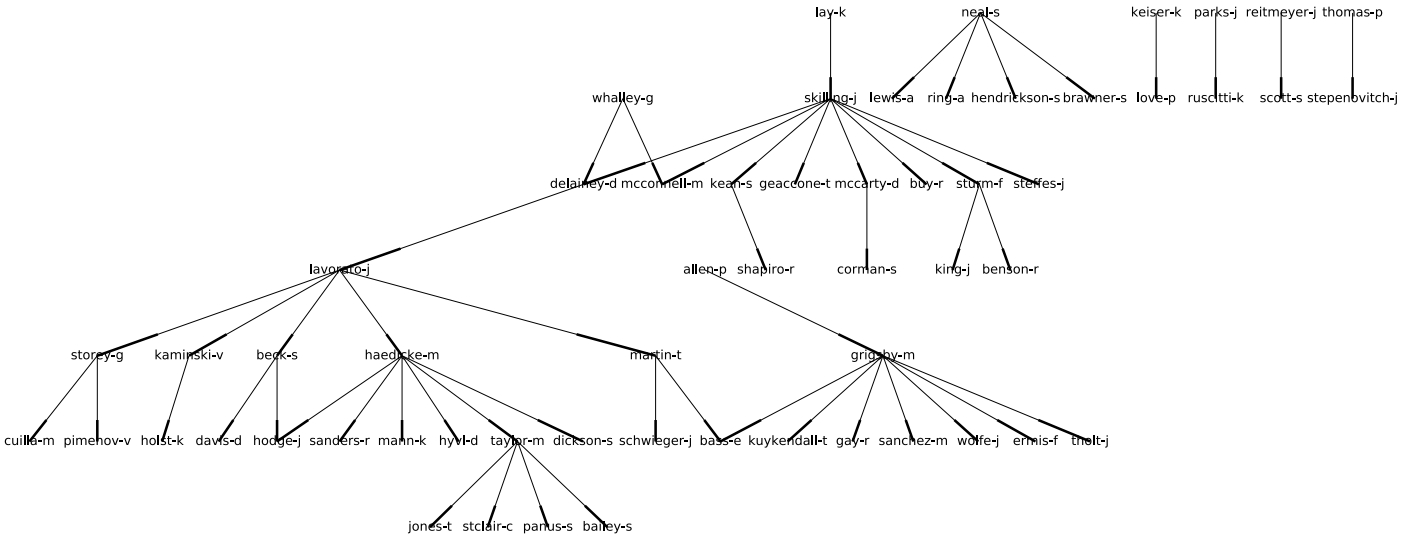


Figure 4: A visual depiction of the hierarchy of Enron job titles.[3]

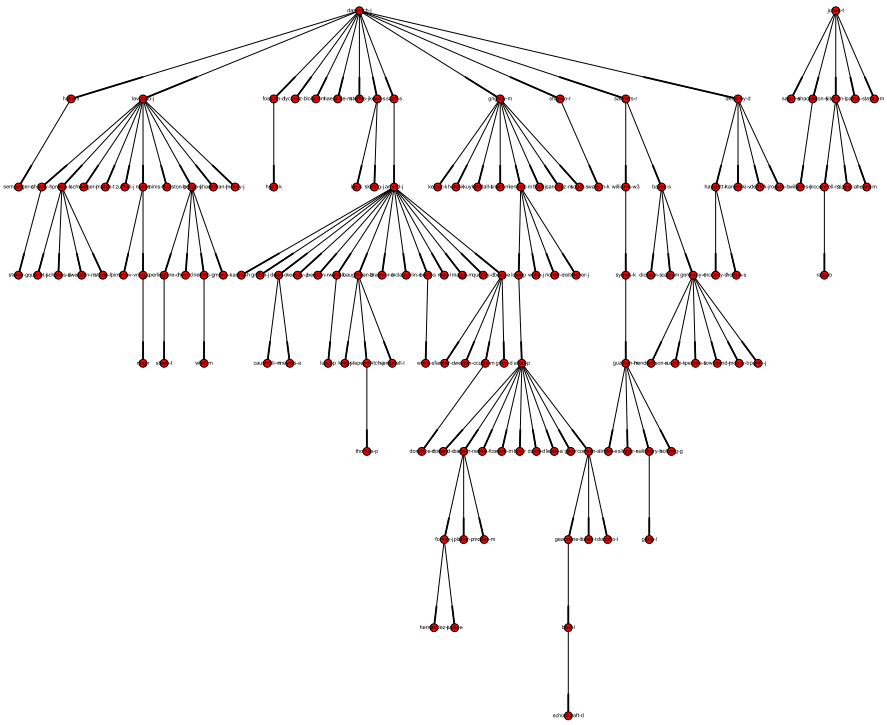


Figure 5: Hierarchy as created by our implementation of Zhou's greedy algorithm, for minimum emails exchanged between two users set to 3, which produced the best Recall