Social Network Analysis of the Enron Corpus

Richard Careaga

April 25, 2019

Introduction

Goal

The goal of this paper is to illustrate techniques of social network analysis in combination with natural language processing to identify discrete email subsets in the Enron Corpus.[1]

The Enron Corpus is a collection of 500,000 emails obtained by the Federal Energy Regulatory Commission in plaintext form for a regulatory investigation, made public pursuant to a Freedom of Information Act request. In major litigation, it is not unusual for comparable volumes of email to be collected and reviewed. The conventional method of review is a keyword search.[2] Inevitably, the large majority of emails are barren of useful information.

Method

The process can be improved by preprocessing emails to be reviewed to construct an internal social network through methods of graph analysis. For this paper a latent cluster random effects model was applied.[3]

To provide an informal test of the efficiacy of the latent graph classification, the vocabulary of each group was compared. Each shared distinct words in common, but each had unshared distinct terms with other groups. One group had a vocabulary with approximately 11.21% distinct words that did not appear in either other cluster.

As a method of reviewing emails, the machine learning approach of this approach has two principal benefits. The minimum information needed, a unique identifier for sender and receiver is either already available or extracted early in the process and so represents no additional effort. Judicious subsetting of users, based on graph metrics of centrality, to reduce the graph size, reduces the most computationally intensive portion of the work, latent model fitting. The second benefit is the ability to prioritize review of emails by graph cluster and with knowledge of the relative positions of the participants in the social network of the organization.

Background

> *In times of political turmoil, events can move from impossible to inevitable without even passing through improbable.* Anatole Kalesky

Enron Corp. and its affiliates were engaged in energy-related businesses, as described in its Annual Report on Form 10-K for the year ended December 31, 2000.

[1] The term *corpus* is used in natural language processing to denote a collection of related text.

[2] *See, e.g.,* Advisory Committee, ESI Checklist, ESI Guidelines, keyword limitations, Sedona Conference, and The Federal Rules of Civil Procedure.

[3] *See* Krivitsky P, Handcock M (2018). *latentnet: Latent Position and ClusterModels for Statistical Networks.* The Statnet Project http://www.statnet.org. R package version 2.9.0, https://CRAN.R-project.org/package=latentnet and Krivitsky PN, Handcock MS (2008). "Fitting position latent cluster models for social networks with latentnet." *Journal of Statistical Software*, *24*(5).

```
*    the transportation of natural gas through pipelines to
markets throughout the United States;

*    the generation, transmission and distribution of
electricity to markets in the northwestern United States;

*    the marketing of natural gas, electricity and other
commodities and related risk management and finance services
worldwide;

*    the development, construction and operation of power
plants, pipelines and other energy related assets worldwide;

*    the delivery and management of energy commodities and
capabilities to end-use retail customers in the industrial
and commercial business sectors; and

*    the development of an intelligent network platform to
provide bandwidth management services and the delivery of
high bandwidth communication applications.
```

As of December 31, 2000, Enron employed approximately 20,600 persons.

For the year ended December 31, 2000, it had operating revenues of $100,789 million, according to the same report, in which it described one of its businesses as

> Enron purchases, markets and delivers natural gas, electricity and other commodities in North America. Customers include independent oil and gas producers, energy- intensive industries, public and investor-owned utility power companies, small independent power producers and local distribution companies. Enron also offers a broad range of price, risk management and financing services including forward contracts, swap agreements and other contractual commitments. Enron's strategy is to enhance the scale, scope, flexibility and speed of its North American energy businesses through developing and acquiring selective assets, securing contractual access to third party assets, forming alliances with customers and utilizing technology such as EnronOnline. With increased liquidity in the marketplace and the success of EnronOnline, Enron believes that it no longer needs to own the same level of physical assets, instead utilizing contracting and market-making activities.

On December 2, 2001, Enron filed for bankruptcy protection.

In less than a year, Enron underwent a complete reversal of fortune as its business strategies ran afoul of applicable regulations, among which were those of the Federal Energy Regulatory Commission (**FERC**).

FERC became aware of irregularities in the California wholesale electricity market prices, a business in which Enron participated. An orientation to the issues is provided by testimony before FERC, which provides a concise summary.[4]

Following Enron's bankruptcy, FERC intensified its investigation, including examining the email records of 149 Enron employees. A preliminary staff report issued six months later.

[4]The short version, which I can relate as a former California electric utility regulatory official from personal knowledge, is that public electric utilities were losing a large share of industrial customers to self-generation. Many businesses found it cheaper to generate on-site than to pay tariff rates. Foreseeably, residential and business customers without the option to self-generate would come to bear the entire cost of unamortized utility fixed assets (termed *stranded costs*), and rates for retail, commercial and small industrial customers would increase. The adopted solution was to require the utilities to sell their generation plants and buy power on a new public market on a *day-ahead*, tomorrow's estimated demand, and an *hour-ahead* basis for unanticipated demand. Although much thought was devoted to the dangers that participants would game the system to sell at premiums or buy at discounts from market, insufficient consideration was given to multi-participant cooperation.

Motivating Data

FERC obtained approximately 500,000 emails. Copies of these were acquired by Leslie Kaelbling of MIT and published by William W. Cohen of Carnegie Mellon University. It is one of the largest publicly available datasets of corporate email and is referred to as the Enron Corpus.

At the time, electronic record examination (*ediscovery*) in litigation was in a primitive state. It was not uncommon, for example, for paper copies of email to be offered. These would typically be read by teams of freelance attorneys looking for keywords. Advanced technology included scanning with optical character recognition and some proprietary software options to organize emails and capture the status of review.

Much of the focus was directed to keyword searches, sometimes called the *smoking gun* approach. Brute force examination misses opportunities to understand the social networks that reflect how the organization operates, what their concerns are and the haphazard exposure of document reviewers inevitably poses the Elephant and the Blind Men Problem. To triage the corpus quickly and efficiently, it should first be distilled and analyzed in terms of its social network characteristics – who corresponds privately with whom.

Analysis

Data acquisition

I obtained a copy of the 2009 version of the corpus in 2010. It contains copies of emails of a private nature that involve three users have who since requested 27 emails to be redacted. I have removed those.[5]

The following were extracted from the SQL database I prepared for my 2010 analysis on the graph portion of this paper.

```
+----------+--------------+------+-----+---------+-------+
| body     | mediumtext   | YES  |     | NULL    |       |
| lastword | mediumtext   | YES  |     | NULL    |       |
| hash     | varchar(250) | YES  | UNI | NULL    |       |
| sender   | varchar(250) | YES  |     | NULL    |       |
| tos      | text         | YES  |     | NULL    |       |
| mid      | varchar(250) | YES  |     | NULL    |       |
| ccs      | text         | YES  |     | NULL    |       |
| date     | datetime     | YES  |     | NULL    |       |
| subj     | varchar(500) | YES  |     | NULL    |       |
| tosctn   | mediumint(9) | YES  |     | NULL    |       |
| ccsctn   | mediumint(9) | YES  |     | NULL    |       |
| source   | varchar(250) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
```

The principal fields used in this paper are:

- sender
- date
- subject
- recipient
- lastword (content in the email that does not occur in its related thread, if any)

[5]Most of my work on data wrangling and preliminary analysis took place in 2010 in Python, relying heavily on the NLTK and networkx packages. For this paper, I did not consult the literature related to graph analysis using the Enron Corpus as an example.

Conversion

Each email was a plaintext file[6] Each user had a directory tree similar to the one below.[7]

Although tedious, traversing the directory tree, parsing the emails and loading them into an SQL database, was accomplished with a combination of Python and Perl scripting and standard bash tools. I do not reproduce that process here as it has little bearing on the main topic of this paper.[8]

Data structure

While the emails were not in native format, the plain text versions contained nine principal segments, as shown in the figure below

Deduplication

Using scripting tools, each text file extraction created a *payload* of the new content in the related email, capturing the text between the beginning metadata and the following metadata for email purposes. A `payload` hash, an md5 encoded message digest[9] was used in the initial analysis as a primary key to assure the uniqueness of each record. Approximately half of the corpus consisted of duplicates, such as the original message in the sender's sent file and one or more copies in the recipient's inbox, at a minimum. Multiple recipients and recipients who used email folders as a filing system were another source of duplicate messages. Applying this filter reduced the corpus to approximately 250,000 emails.

Text isolation

For natural language processing (**NLP**) purposes, treating the `payload` rather than the `message body` as the unit of analysis avoided an *echo chamber* effect of `chains` quoting and re-quoting the original message, multiplying the frequency of the words it contained.

Prioritization

Traditional litigation analysis of emails was conducted on the principle that *something may be overlooked,* which delays the value of email in preliminary analysis. Prioritizing always leaves open the option of reviewing the set-asides later.

After deduplication, the first filter applied was to eliminate all email from external addresses that were not also recipients from internal addresses. Spam, newsletters and the like have low information potential. This filter reduced the remaining half of the corpus by half again, leaving approximately 125,000 emails.

A second filter for internal email was used to eliminate broadcast messages and high frequency administrative messages. Indicia of broadcast messages were large numbers of recipients, high frequency, paucity of return correspondence and keyword in context screening. Administrative messages to single recipients were identified by frequency, lack of return correspondence and high frequency words. Many of these were nagging emails concerning the lack of approval of expense reports, for example. This filter reduced the dataset to approximately 35,000.

[6]Most had been generated by Microsoft Outlook, but some older emails were produced in IBM Notes, which created some character encoding issues.

[7]This user had 10 directories with 3048 files (the directory tree illustration has been pruned to omit spurious detail) containing 12,147 lines and 69,226 words.

[8]For this paper, supplemental processing of the recipient field was necessary and reflected in the script to remove spurious punctuation, such as the newline character embedded as slash-n.

[9]In theory, it is possible that two non-identical sequences of bytes be encoded identically; the probability is low enough to make an md5 digest usable as a checksum verification, its purpose here.

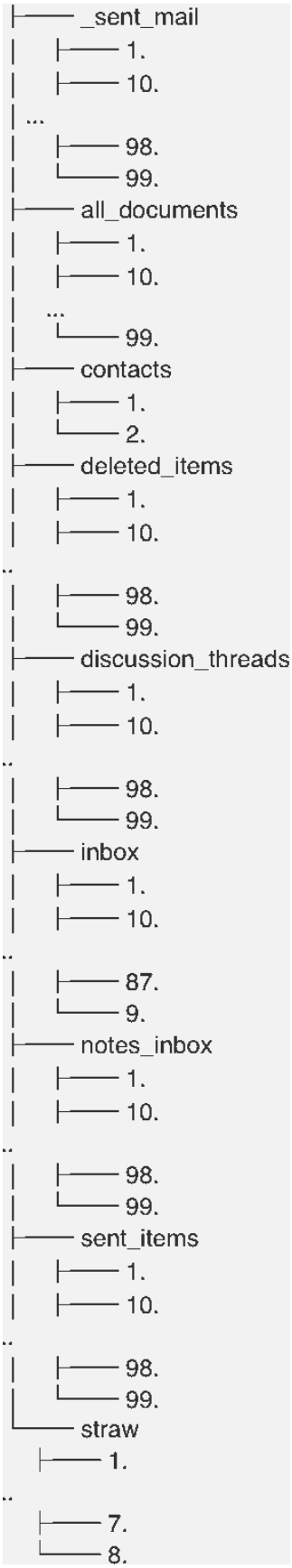


Figure 1: Typical user data

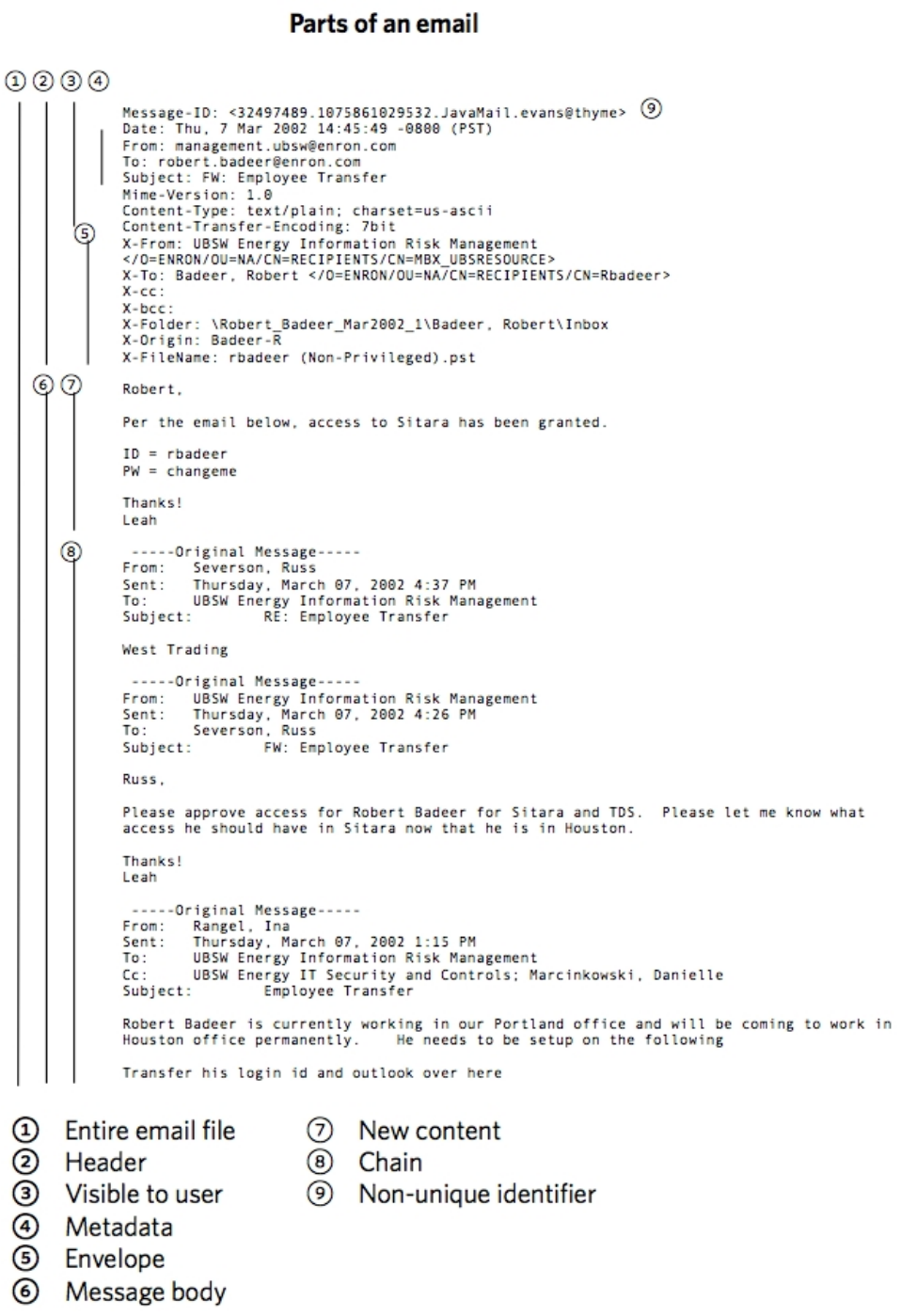


Figure 2: Structural analysis of an Enron email

The third filter limited the dataset to emails sent before Enron's December 2, 2001 bankruptcy. This filter reduced the email count to approximately 13,500, about 2.7% of the original total. A few emails dated "1979-12-31" were reviewed and deleted. The resulting dataset was named `g_enron` for its initial purpose, network graph analysis.

Social network analysis

The nature of social networks

Following the reduction of the corpus, the remaining senders and receivers were natural persons who engaged in mutual correspondence. These constitute `nodes` or `vertices` and their emails `edges`[10]. Draw three points and connect them, and you have created three nodes and three edges, a triangle, which is termed a `triad graph` object. A graph object encapsulates many useful features aside from who knows whom[11], including measures of density, centrality, connectedness, separation, clustering and other indicia of how well or poorly embedded in an organization any individual may stand.

Graphs are potentially computationally intensive, which motivated the initial reduction of the selection of emails and users to approximately 1.4% of the emails available for examination. In addition, moving from bigraph directed network to multidirected graph[12] was infeasible.

Graphs are not only a processing unit, they constitute the domain of their own branch of mathematics.[13].

Augmentation and transformation

Each unique Enron address in the reduced dataset was assigned a userid. The primary purpose was to facilitate social network analysis with node identifiers of uniform length; the second, to reduce analyst bias arising from gender stereotyping, frequency of exposure and similar subjective pattern seeking behaviors.

To achieve a computationally practicable dataset for initial social network analysis, emails were limited to single Enron sender to Enron single recipient, reducing the dataset further, to 7,884 emails.

The network composition

Time frame

All emails from January 1, 2000 to December 2, 2001 the date of the bankruptcy were collected. A handful of messages prior to January 1, 2000 were excluded due to their low counts.

Users

A total of 2,111 unique users are represented. However, all but 1107 users are non-reciprocating or isolated. To identify those, the sender and recipient userids were extracted and converted to a graph object, which will be referred to as the **reduced Enron corpus.** Its attributes are

```
##  Network attributes:
##   vertices = 91
##   directed = TRUE
```

[10] Or `arcs`, when directionality is considered

[11] Such as the parlor game six degrees of Kevin Bacon

[12] A multidirected graph has a single edge to multiple vertices; the analysis is beyond the scope of a term paper for a network as large as the Enron Corpus.

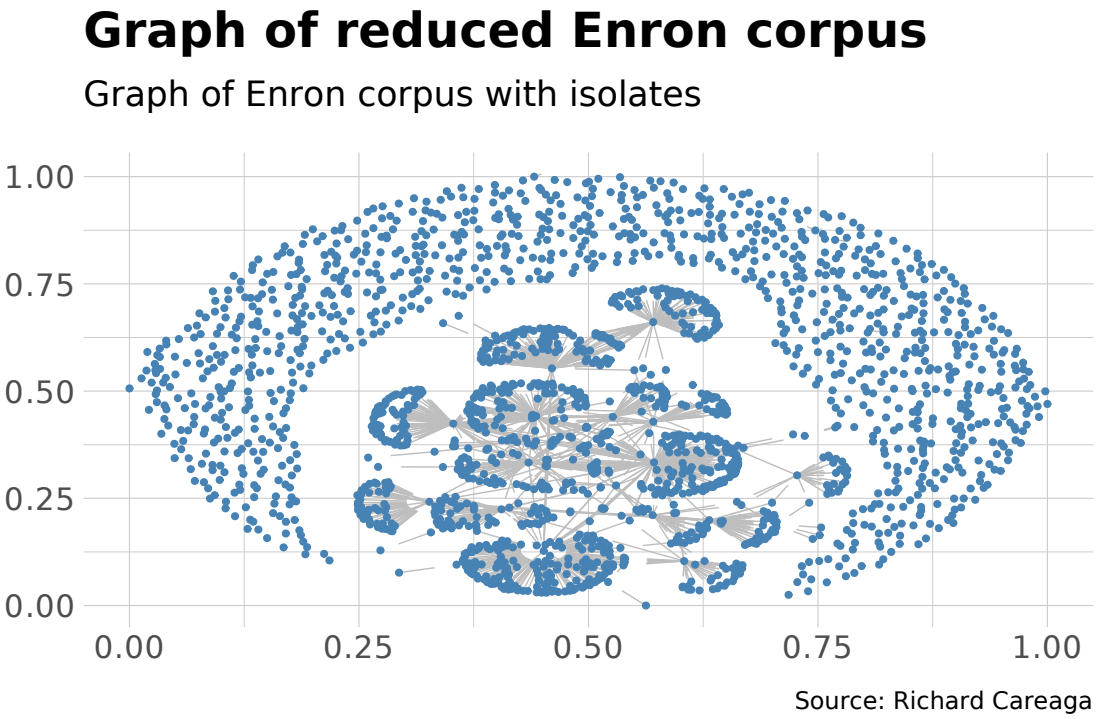
[13] *See, e.g.*, the brief tutorial by Keijo Ruohonen

```
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 1281
##     missing edges= 0
##     non-missing edges= 1281
##
##  Vertex attribute names:
##     sts vertex.names
##
##  Edge attribute names not shown
```

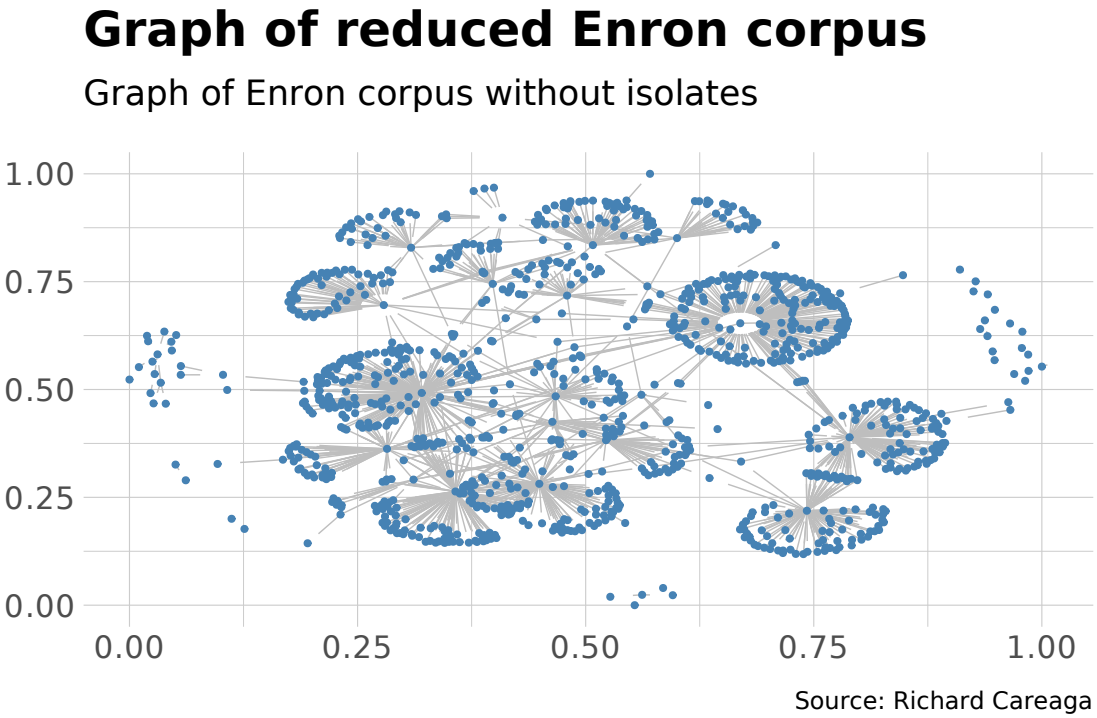
Definition of terms:

- vertices: users
- directed: from-to and to-from distinguished
- hyper: contains emails from or to multiple users
- loops: includes email from user to herself
- multiple: multi-dimensional object
- bipartite: set of two vertices where no vertex in the same set is connected
- edges: number of emails

The graph can be visualized in several ways. Here, and throughout the paper, a representation based on the Fructerman-Reingold force-directed algorithm[14] is used to promote visual discrimination.



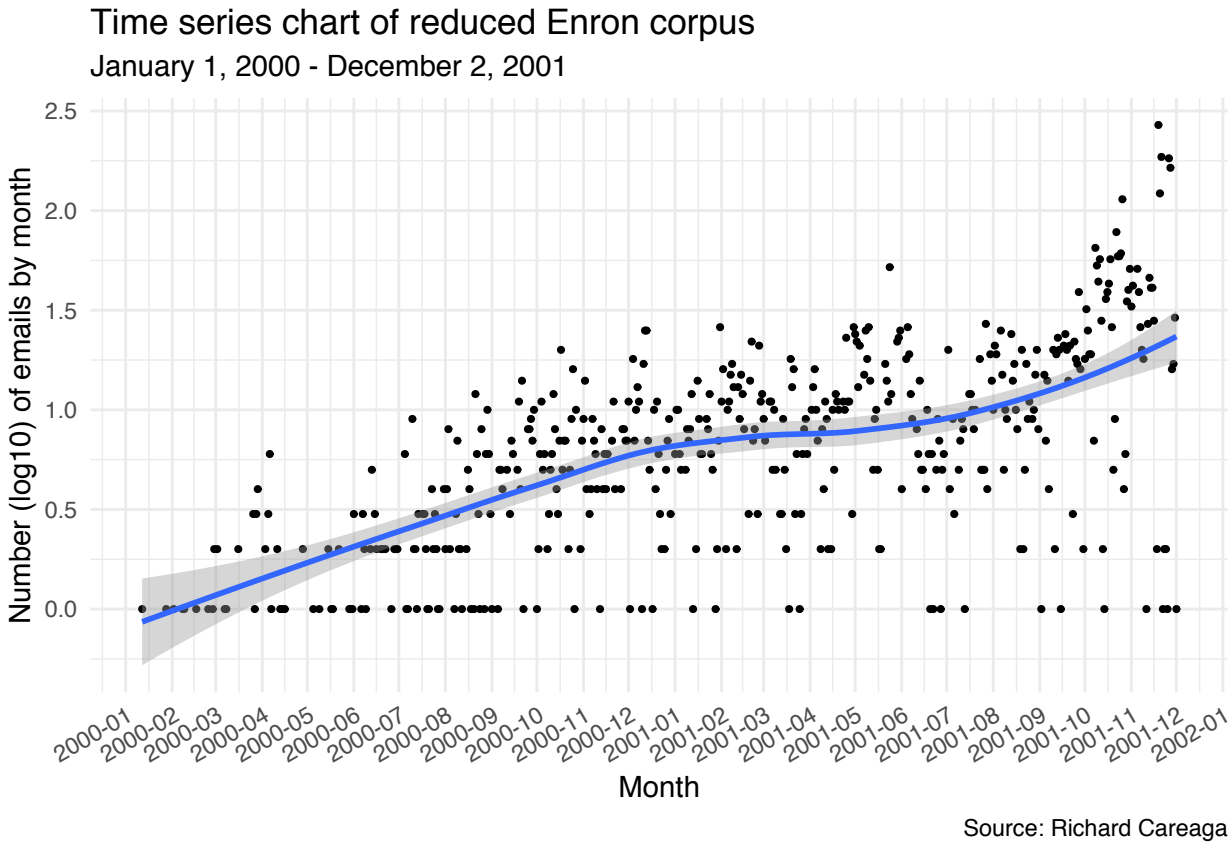
[14]Fruchterman, T. M. and Reingold, E. M. (1991), Graph drawing by force-directed placement. Softw: Pract. Exper., 21: 1129-1164. doi:10.1002/spe.4380211102



Graph objects shown here represent users (vertices) by dots and emails (edges) by lines. The length of the line is not a measure of distance. The visualization algorithm arranges vertices and edges to promote recognition of connections only.

Time series of reduced Enron corpus January 2000-December 2001

Several groups of outliers are apparent, notably mid-May 2001 and the weeks leading up to the bankruptcy.



Transitivity

Graph transitivity is a measure of the likelihood that two pairs of vertices (*dyads*) are likely to be strongly connected $A- > B- > C => A- > C$ in the weak form and $A- > B- > C <=> A- > C$ in the strong form. The `sna:gtrans` strong form measure for the graph is 0.9987

User prominence

Graph measures of user prominence

All of the functions described in this section[15], `degree`, `loadcent` and `stresscent` have been run with the `rescale = TRUE` option to normalize them. The functions measure the prominent of a vertex in different ways. The `sna::degree` function relies on measures of incoming and outgoing connections. The `sna::loadcent` function

> measures the degree to which a vertex is in a position of brokerage by summing up the fractions of shortest paths between other pairs of vertices that pass through it. Brandes[16]

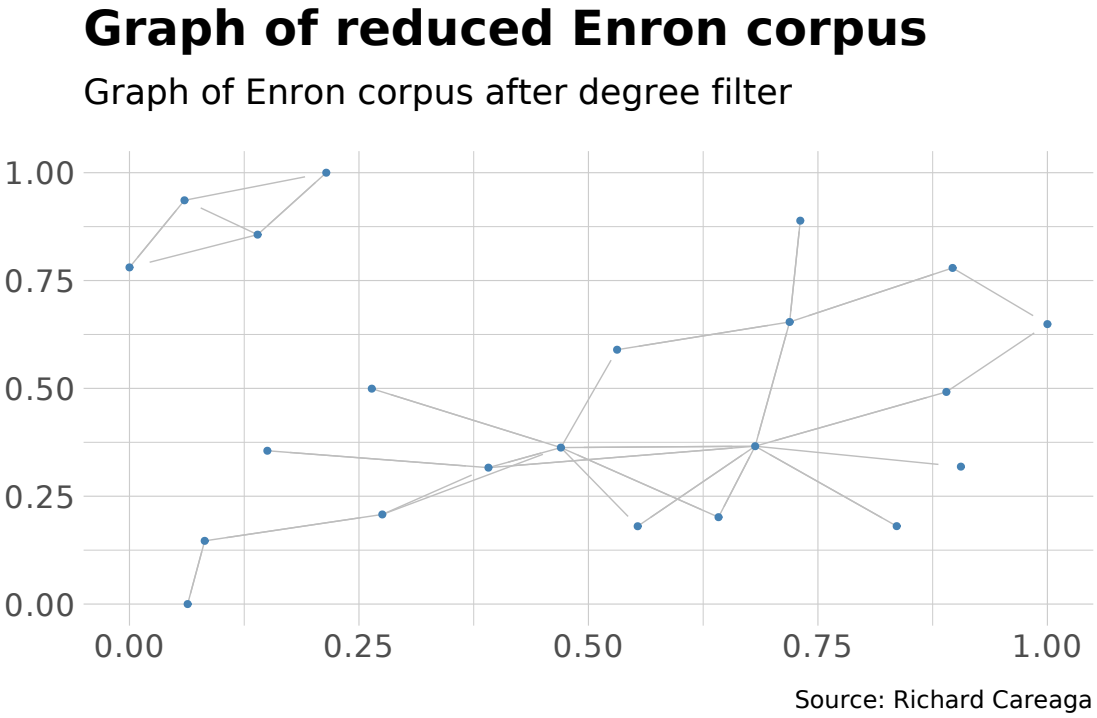
The `sna::stresscent` function is a measure of the shortest number of edges that a vertex has to traverse to reach every other vertex in a graph.

[15]Rejected measures of graph centrality of vertices: betweenness (redunant with ldctr); infocent (all 1.206801e-13); closeness (all 0); evcent (asymmetry failure); bonpow (system is exactly singular error); flowbet(ran without finishing); graphcent (all 0)

[16]Brandes, U. (2008). "On Variants of Shortest-Path Betweenness Centrality and their Generic Computation." Social Networks, 30, 136-145.

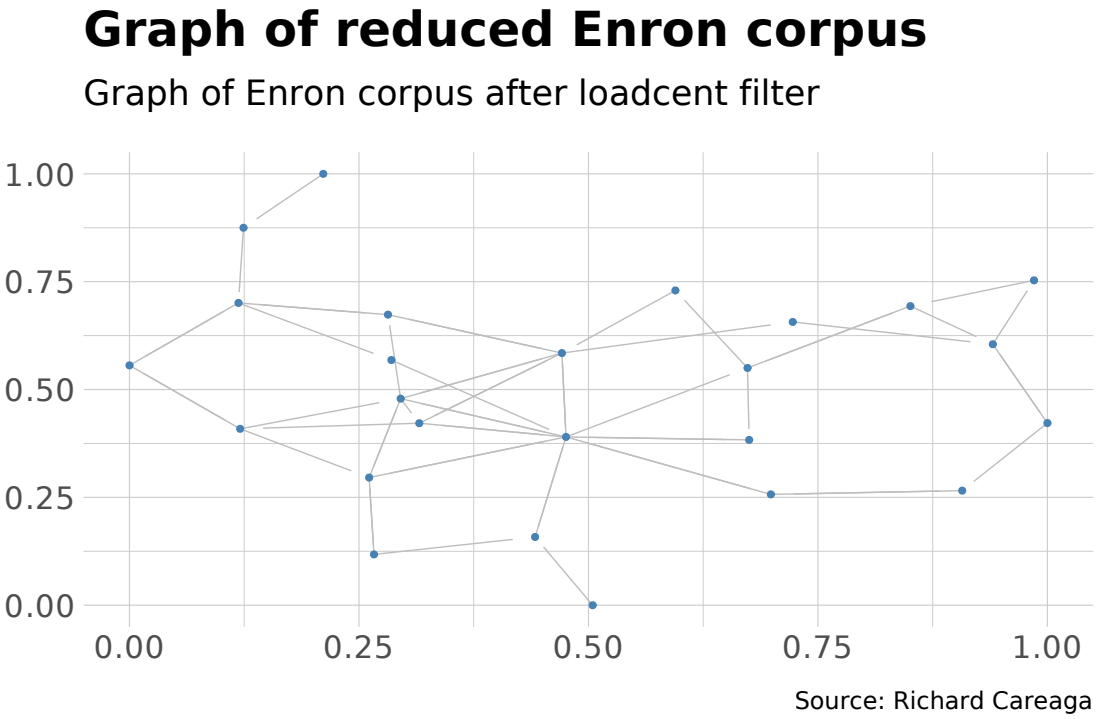
Degree

A graph of the top 25 users ranked by `degree` as a sender or receiver is shown below.



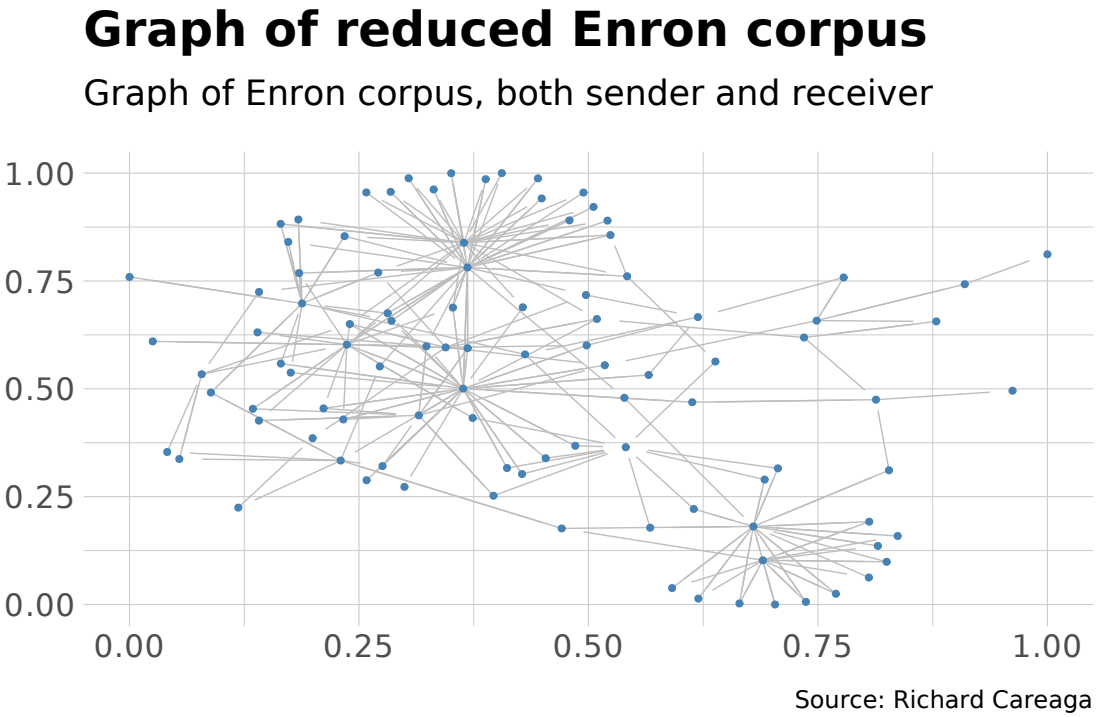
Load centrality

A graph of the top 25 users ranked by `load centrality` as a sender or receiver is shown below.



Stress centrality

A graph of the top 100 users ranked by ‘stress centrality users as a sender or receiver is shown below.



Usefulness of combination measures

Which of these measures to privilege is not clear. They each present different perspectives of the relative importance of each user in the network, based on different criteria, but none presents an obvious candidate by itself. They are, however, moderately well correlated at high degrees of significance.

Table 1: Pearson’s product-moment correlation: `deg` and `ldctr`

Test statistic	df	P value	Alternative hypothesis	cor
30.71	1105	2.905e-150 * * *	two.sided	0.6786

Table 2: Pearson’s product-moment correlation: `deg` and `sts`

Test statistic	df	P value	Alternative hypothesis	cor
57.47	1105	0 * * *	two.sided	0.8656

Table 3: Pearson’s product-moment correlation: `ldctr` and `sts`

Test statistic	df	P value	Alternative hypothesis	cor
34.26	1105	7.481e-176 * * *	two.sided	0.7177

The union and intersection of the top 25 users using each centrality measure were identified and rejected

because their use in subsequent latent network identification either failed or produced unfavorable diagnostics.

Stress centrality was selected because it correlated best with the other two methods and produced satisfactory latent network results as discussed below.

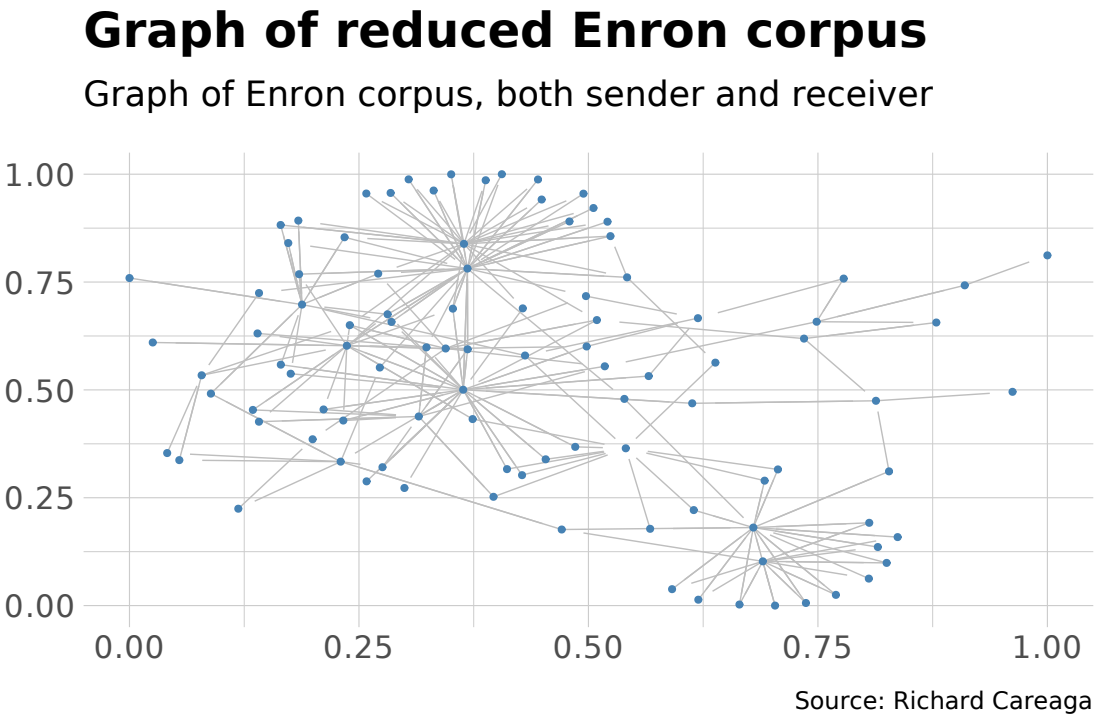
Latent network analysis

Using the top 100 `stresscent` users (rather than the top 25) yields 2,349 edges, each representing an email.

Two latent network analyses were performed. The first selected users who were among the top 100 if they appeared *both* as sender and recipient. The second selected users who were among the top 100 *either* as sender or receiver; that model failed to complete within two hours and was discarded.

The latent cluster random model of senders and receivers

The graph object, prior to modeling, appeared as follows:



The `latent::ergmm` model was applied to the graph.

```
c.fit <- ergmm(net_c ~ euclidean(d=2, G=3)+rreceiver,
      control=ergmm.control(store.burnin=TRUE), seed = 2203)
```

The function fits the graph to a latent network model using a Markov chain Monte Carlo algorithm for a Bayesian model fit. The resulting graph visualization identifieds three clusters. Some vertices show pie slices indicating the relative probabilities of belonging to one of the three clusters. The diagnostics include an intercept estimate, confidence intervals, and a p-value, all of which are satisfactory.

```
##
## ==========================
## Summary of model fit
## ==========================
```

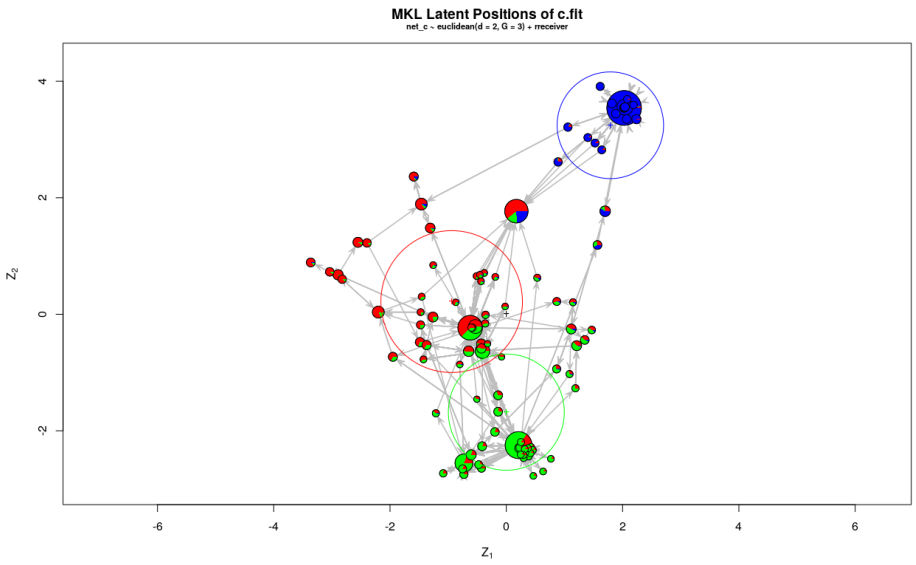


Figure 3: Latent graph based on stresscent

```
##
## Formula:   net_c ~ euclidean(d = 2, G = 3) + rreceiver
## Attribute: edges
## Model:     Bernoulli
## MCMC sample of size 4000, draws are 10 iterations apart, after burnin of 10000 iterations.
## Covariate coefficients posterior means:
##             Estimate     2.5%   97.5% 2*min(Pr(>0),Pr(<0))
## (Intercept) -0.76477 -1.15728 -0.5002            < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Receiver effect variance: 0.8133433.
## Overall BIC:        2464.52
## Likelihood BIC:     1625.131
## Latent space/clustering BIC:     648.8901
## Receiver effect BIC:     190.4993
##
## Covariate coefficients MKL:
##              Estimate
## (Intercept) -1.819249
```

Convergence of the model is shown by diagnostic plots of autocorrelation. The log probability (`lpY`) decreases, as does the probability vector (`beta.1`), and the receiver random effect (`receiver1`). The point estimates Z.1.1 and Z.1.2 are consistent across lags. The following traces and densities unskewed distributions, and the goodness of fit plots are reasonable.

Goodness of fit diagnostics for in-degree, out-degree and geodesic distance are provided in tabular format and plots. Some excursions in each of the plots appear, indicating the potential benefit of further model tuning.

```
##
## Goodness-of-fit for in-degree
```

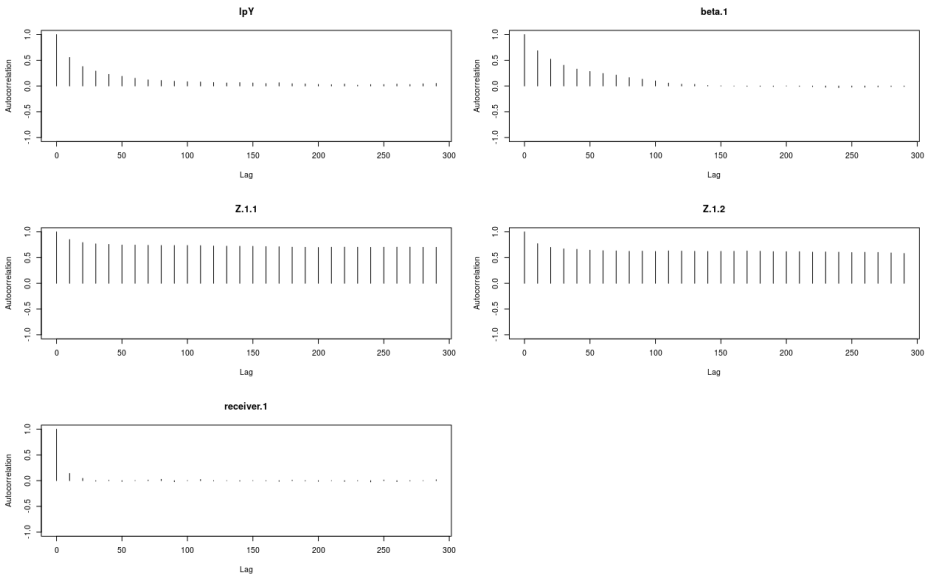


Figure 4: Fit diagnosis part 1

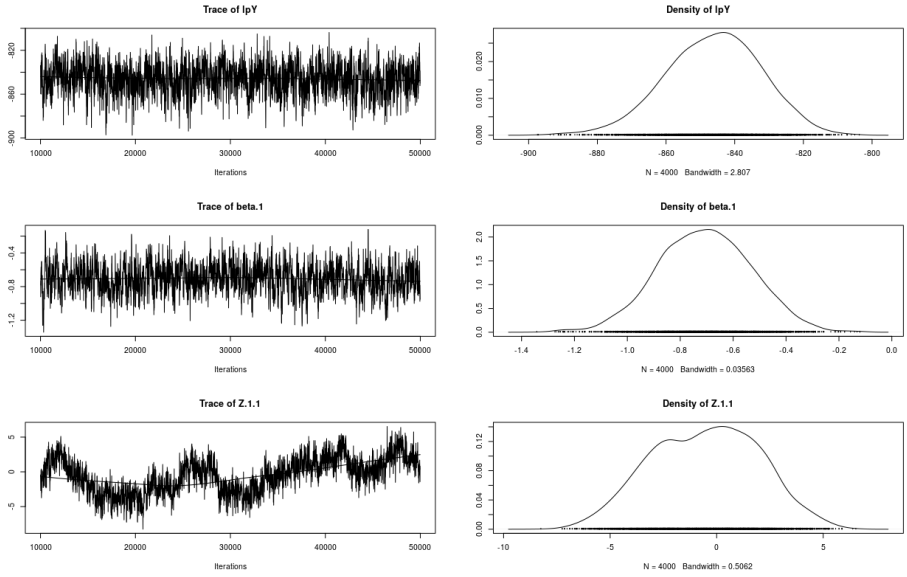


Figure 5: Fit diagnosis part 2

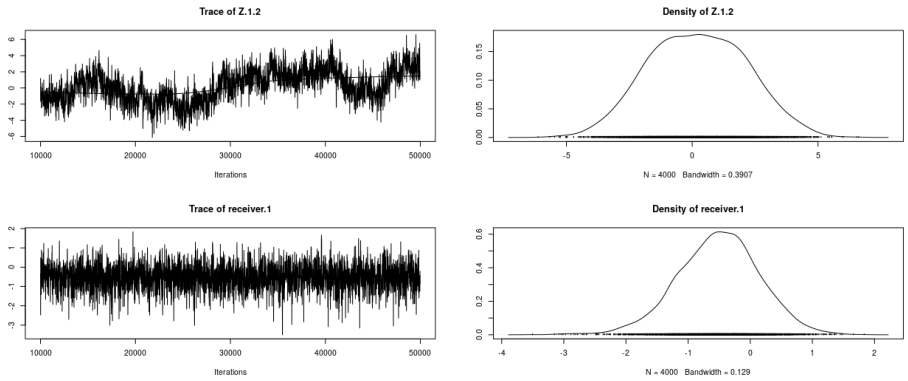


Figure 6: Fit diagnosis part 3

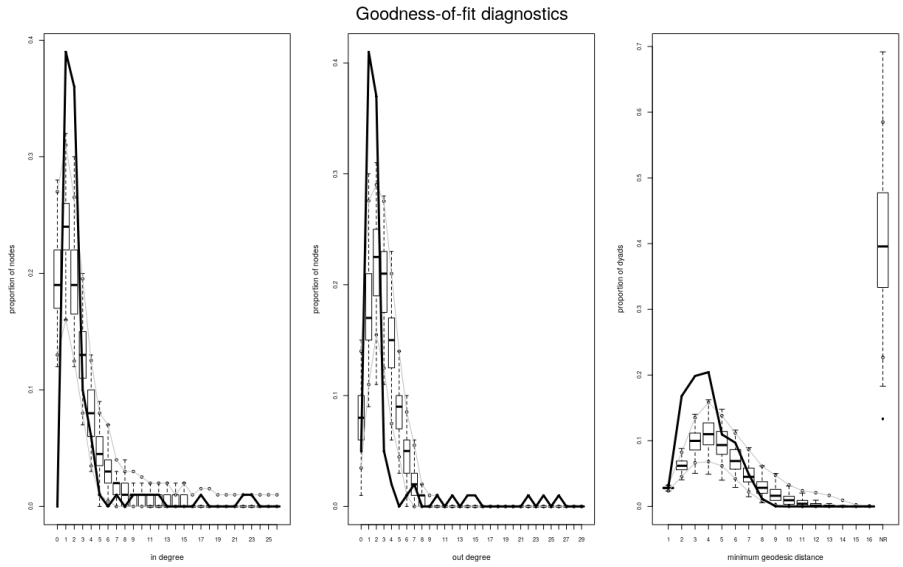


Figure 7: Goodness of fit part 1

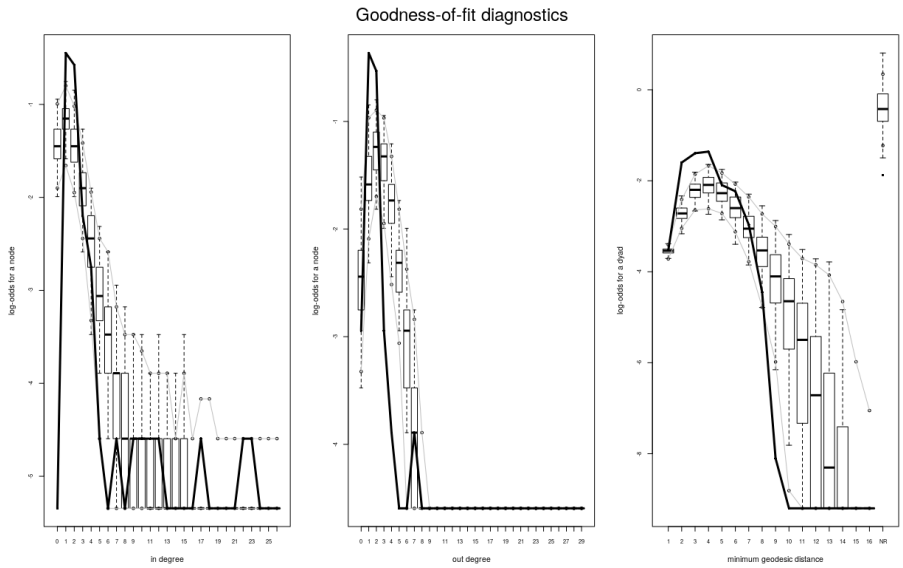


Figure 8: Goodness of fit part 2

```
##
##    obs min  mean max MC p-value
## 0    0  11 19.95  30       0.00
## 1   39  14 23.33  33       0.00
## 2   36  11 18.40  26       0.00
## 3   10   6 13.10  21       0.40
## 4    6   2  8.14  16       0.60
## 5    1   1  5.28  11       0.02
## 6    0   0  3.04   8       0.16
## 7    1   0  1.53   5       1.00
## 8    0   0  1.42   5       0.52
## 9    1   0  0.88   4       1.00
## 10   1   0  0.77   3       1.00
## 11   1   0  0.58   2       0.92
## 12   1   0  0.43   3       0.70
## 13   0   0  0.46   2       1.00
## 14   0   0  0.36   3       1.00
## 15   0   0  0.34   3       1.00
## 16   0   0  0.27   2       1.00
## 17   1   0  0.20   2       0.36
## 18   0   0  0.29   2       1.00
## 19   0   0  0.19   2       1.00
## 20   0   0  0.20   2       1.00
## 21   0   0  0.15   1       1.00
## 22   1   0  0.14   1       0.28
## 23   1   0  0.12   1       0.24
## 24   0   0  0.20   2       1.00
## 25   0   0  0.07   1       1.00
## 26   0   0  0.05   1       1.00
## 27   0   0  0.05   1       1.00
## 28   0   0  0.01   1       1.00
## 30   0   0  0.03   1       1.00
```

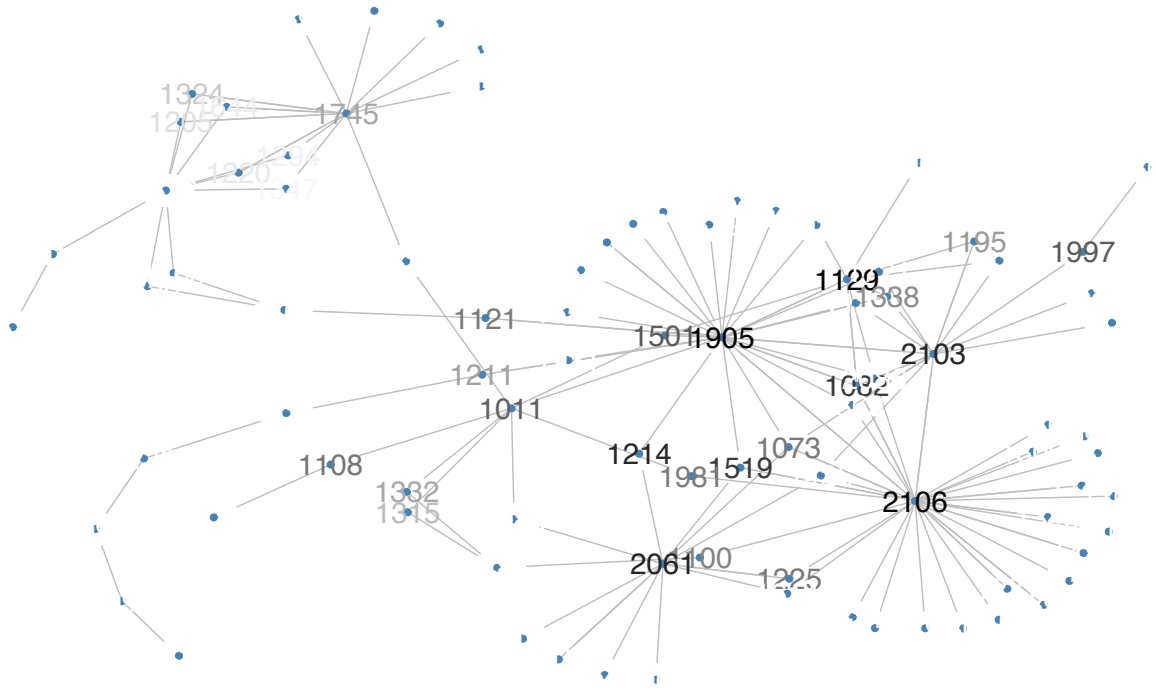
```
## 32   0   0  0.01   1       1.00
## 33   0   0  0.01   1       1.00
##
## Goodness-of-fit for out-degree
##
##    obs min  mean max MC p-value
## 0    5   3  8.32  19       0.28
## 1   41   8 17.25  28       0.00
## 2   37  13 21.77  33       0.00
## 3    5   9 19.48  31       0.00
## 4    2   8 15.57  24       0.00
## 5    0   4  9.44  17       0.00
## 6    1   1  4.98  11       0.12
## 7    2   0  2.04   6       1.00
## 8    0   0  0.80   3       0.84
## 9    0   0  0.27   2       1.00
## 10   1   0  0.05   1       0.10
## 11   0   0  0.01   1       1.00
## 12   1   0  0.02   1       0.04
## 14   1   0  0.00   0       0.00
## 15   1   0  0.00   0       0.00
## 22   1   0  0.00   0       0.00
## 24   1   0  0.00   0       0.00
## 26   1   0  0.00   0       0.00
##
## Goodness-of-fit for minimum geodesic distance
##
##      obs  min    mean  max MC p-value
## 1    281  244  282.55  317       0.86
## 2   1663  423  631.63  857       0.00
## 3   1966  613  992.83 1385       0.00
## 4   2023  750 1112.23 1709       0.00
## 5   1085  621  963.09 1405       0.56
## 6    960  381  716.66 1182       0.22
## 7    486  156  483.25  838       0.94
## 8    114   42  303.05  648       0.14
## 9      3    6  178.62  482       0.00
## 10     0    0   97.71  365       0.02
## 11     0    0   50.07  263       0.24
## 12     0    0   25.34  276       0.54
## 13     0    0   13.75  271       1.00
## 14     0    0    8.19  274       1.00
## 15     0    0    4.10  176       1.00
## 16     0    0    1.56   78       1.00
## 17     0    0    0.50   26       1.00
## 18     0    0    0.11    6       1.00
## 19     0    0    0.02    2       1.00
## 20     0    0    0.01    1       1.00
## Inf 1319 2262 4034.73 6162       0.00
```

The three clusters can be examined separately.[17]

[17] Highlighted vertices are those included in the top 100 `stresscent` group.

Graph of reduced Enron corpus

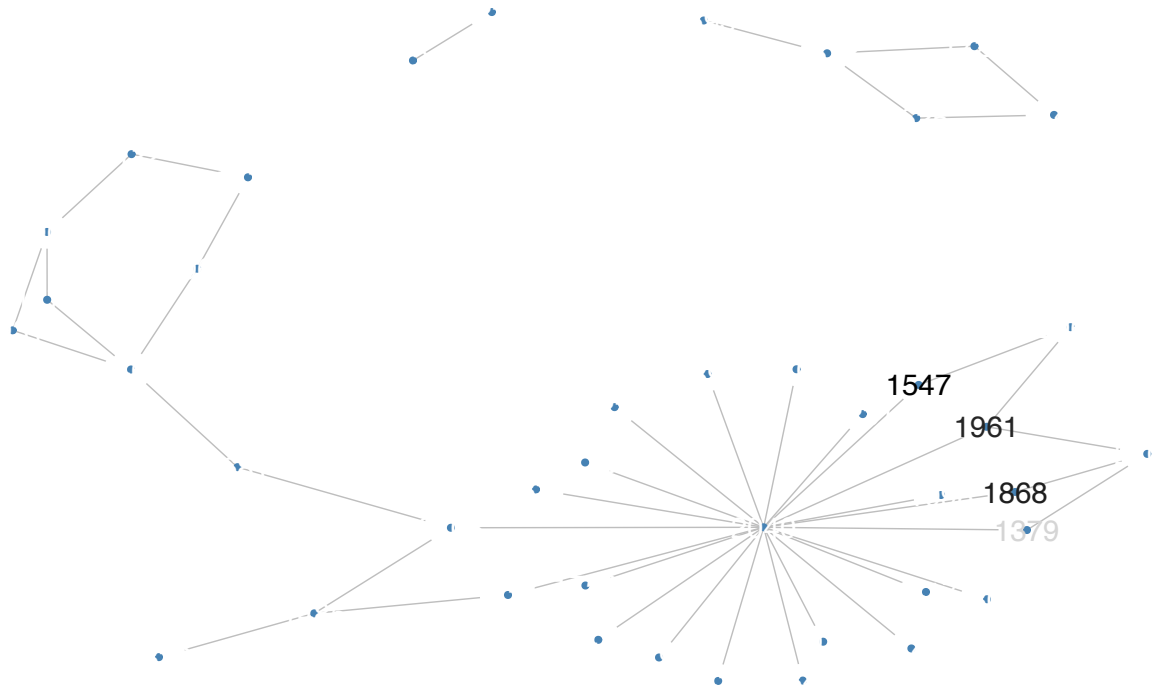
Cluster 1



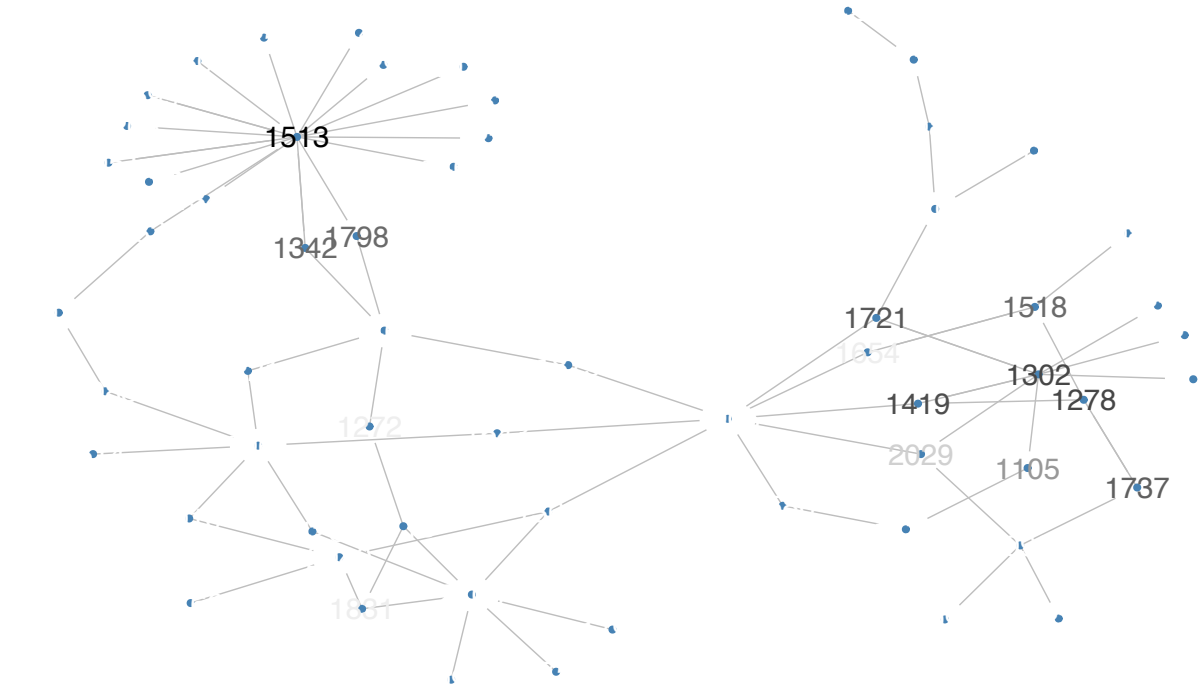
Source: Richard Careaga

Graph of reduced Enron corpus

Cluster 2



Source: Richard Careaga



Source: Richard Careaga

Clusters 1 and 3 have the greatest number of vertices and edges. Six distinct userids stand out. Cluster 3 has a relative paucity.

Results

Distinct social networks were identified through latent cluster random graph methods. As a by-product, prominent individuals in the network were identified.

Aside from the visually prominent vertices in the plots above, a simple word frequency analysis of two of the clusters displays markedly differing vocabularies. The third cluster contains no unique terms. The clusters are a subset (based on high `stresscent` scores of the larger corpus) that has an added field for cluster membership.

Within those clusters are 6,733 distinct words. Of those, 27.71% are unique to Cluster 1;
11.21% are unique to Cluster 2; and 0% are unique to Cluster 3.

Conclusion

The hypothesis of this paper is that social network analysis preprocessing of email text is a feasible method to rapidly identify users who form subgroups with email content of potential interest. Relying solely on metadata (sender/receiver), latent network analysis identified three sub-graphs that have distinct vocabularies.

Credits

Simon Urbanek and Jeffrey Horner (2019). Cairo: R Graphics Device using Cairo Graphics Library for Creating High-Quality Bitmap (PNG, JPEG, TIFF), Vector (PDF, SVG, PostScript) and Display (X11 and Win32) Output. R package version 1.5-10. https://CRAN.R-project.org/package=Cairo

Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines (2006). CODA: Convergence Diagnosis and Output Analysis for MCMC, R News, vol 6, 7-11

Winston Chang, (2014). extrafont: Tools for using fonts. R package version 0.17. https://CRAN.R-project.org/package=extrafont

Francois Briatte (2016). ggnetwork: Geometries to Plot Networks with 'ggplot2'. R package version 0.5.1. https://CRAN.R-project.org/package=ggnetwork

Kirill Müller (2017). here: A Simpler Way to Find Your Files. R package version 0.1. https://CRAN.R-project.org/package=here

Handcock M, Hunter D, Butts C, Goodreau S, Krivitsky P, Morris M (2018). *ergm: Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project http://www.statnet.org. R package version 3.9.4, https://CRAN.R-project.org/package=ergm

Hunter D, Handcock M, Butts C, Goodreau S, Morris M (2008). "ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks." *Journal of Statistical Software*, *24*(3), 1-29.

Barret Schloerke, Jason Crowley, Di Cook, Francois Briatte, Moritz Marbach, Edwin Thoen, Amos Elberg and Joseph Larmarange (2018). GGally: Extension to 'ggplot2'. R package version 1.4.0. https://CRAN.R-project.org/package=GGally

Francois Briatte (2016). ggnetwork: Geometries to Plot Networks with 'ggplot2'. R package version 0.5.1. https://CRAN.R-project.org/package=ggnetwork

Jeffrey B. Arnold (2019). ggthemes: Extra Themes, Scales and Geoms for 'ggplot2'. R package version 4.1.1. https://CRAN.R-project.org/package=ggthemes

Bob Rudis (2019). hrbrthemes: Additional Themes, Theme Components and Utilities for 'ggplot2'. R package version 0.6.0. https://CRAN.R-project.org/package=hrbrthemes

Yihui Xie (2019). knitr: A General-Purpose Package for Dynamic Report Generation in R. R package version 1.22. Yihui Xie (2015) Dynamic Documents with R and knitr. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963 Yihui Xie (2014) knitr: A Comprehensive Tool for Reproducible Research in R. In Victoria Stodden, Friedrich Leisch and Roger D. Peng, editors, Implementing Reproducible Computational Research. Chapman and Hall/CRC. ISBN 978-1466561595

Krivitsky P, Handcock M (2018). *latentnet: Latent Position and Cluster Models for Statistical Networks*. The Statnet Project http://www.statnet.org. R package version 2.9.0, https://CRAN.R-project.org/package=latentnet

Krivitsky PN, Handcock MS (2008). "Fitting position latent cluster models for social networks with latentnet." *Journal of Statistical Software*, *24*(5).

Butts C (2015). *network: Classes for Relational Data*. The Statnet Project http://www.statnet.org R package version 1.13.0.1, https://CRAN.R-project.org/package=network

Butts C (2008). "network: a Package for Managing Relational Data in R." *Journal of Statistical Software*, *24*(2). http://www.jstatsoft.org/v24/i02/paper

Gergely Daróczi and Roman Tsegelskyi (2018). pander: An R 'Pandoc' Writer. R package version 0.6.3. https://CRAN.R-project.org/package=pander

Carter T. Butts (2016). sna: Tools for Social Network Analysis. R package version 2.4. https://CRAN.R-project.org/package=sna

Krivitsky P (2019). *statnet.common: Common R Scripts and Utilities Used by the Statnet Project Software*. The Statnet Project http://www.statnet.org. R package version 4.2.0, https://CRAN.R-project.org/package=statnet.common

Silge J, Robinson D (2016). "tidytext: Text Mining and Analysis Using Tidy Data Principles in R." *JOSS*, *1*(3). doi: 10.21105/joss.00037 http://doi.org/10.21105/joss.00037, http://dx.doi.org/10.21105/joss.00037

Hadley Wickham (2017). tidyverse: Easily Install and Load the 'Tidyverse'. R package version 1.2.1. https://CRAN.R-project.org/package=tidyverse

Session Information

```
## R version 3.6.0 (2019-04-26)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Mojave 10.14.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] forcats_0.4.0         stringr_1.4.0         dplyr_0.8.1
##  [4] purrr_0.3.2           readr_1.3.1           tidyr_0.8.3
##  [7] tibble_2.1.2          tidyverse_1.2.1       tidytext_0.2.0
## [10] statnet_2018.10       tsna_0.3.0            ergm.count_3.4.0
## [13] tergm_3.6.0           networkDynamic_0.10.0 sna_2.4
## [16] pander_0.6.3          latentnet_2.9.0       statnet.common_4.3.0
## [19] knitr_1.23            hrbrthemes_0.6.0      ggthemes_4.2.0
## [22] GGally_1.4.0          ergm_3.10.1           network_1.15
## [25] here_0.1              ggnetwork_0.5.1       ggplot2_3.1.1
## [28] extrafont_0.17        Cairo_1.5-10          coda_0.19-2
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.0          jsonlite_1.6        modelr_0.1.4
##  [4] assertthat_0.2.1    cellranger_1.1.0    yaml_2.2.0
##  [7] robustbase_0.93-5   ggrepel_0.8.1       gdtools_0.1.8
## [10] Rttf2pt1_1.3.7      pillar_1.4.1        backports_1.1.4
## [13] lattice_0.20-38     glue_1.3.1          extrafontdb_1.0
## [16] digest_0.6.19       RColorBrewer_1.1-2  rvest_0.3.4
## [19] colorspace_1.4-1    htmltools_0.3.6     Matrix_1.2-17
## [22] plyr_1.8.4          lpSolve_5.6.13.1    pkgconfig_2.0.2
## [25] broom_0.5.2         haven_2.1.0         mvtnorm_1.0-10
## [28] scales_1.0.0        generics_0.0.2      withr_2.1.2
## [31] lazyeval_0.2.2      cli_1.1.0           readxl_1.3.1
## [34] magrittr_1.5        crayon_1.3.4        evaluate_0.14
## [37] tokenizers_0.2.1    janeaustenr_0.1.5   nlme_3.1-140
## [40] MASS_7.3-51.4       SnowballC_0.6.0     xml2_1.2.0
## [43] tools_3.6.0         hms_0.4.2           trust_0.1-7
```

```
## [46] munsell_0.5.0      compiler_3.6.0     rlang_0.3.4
## [49] grid_3.6.0         rstudioapi_0.10    labeling_0.3
## [52] rmarkdown_1.13     gtable_0.3.0       reshape_0.8.8
## [55] R6_2.4.0           lubridate_1.7.4    rprojroot_1.3-2
## [58] stringi_1.4.3      parallel_3.6.0     Rcpp_1.0.1
## [61] DEoptimR_1.0-8     tidyselect_0.2.5   xfun_0.7
```

Author contact

Richard Careaga
public@careaga.net
@technocrat
PO Box 3325
Kirkland, WA 98083