Enron Discovery Revisited: AI-Assisted Detection of Market Manipulation through Community Detection, Latent Space Models, and Temporal Dynamics

Richard Careaga[1]

February 2026

Abstract

This report presents a forensic analysis of the Enron email corpus, utilizing advanced social network analysis (SNA) and natural language processing to identify structural and linguistic signatures of market manipulation during the California electricity crisis of 2000–2001. By applying Louvain community detection and adjacency spectral embedding to a cleaned network of 2,352 internal nodes and 4,510 edges, the study isolates specific organizational sub-groups and communication channels responsible for devising and executing manipulative trading strategies.

The analysis identifies "West Power Trading" as a distinct latent cluster that functionally did not exist prior to the crisis, activating solely to facilitate California operations. Temporal analysis reveals that message volume within this group surged from zero to over 2,000 messages during the crisis window, collapsing immediately post-crisis. Furthermore, the study detects a "Hub/Coordinator" cluster—comprising legal, government affairs, and trading personnel—that acted as the primary bridge for sensitive information flow between operational desks and senior management. Lexical analysis confirms that explicit discussion of "manipulation" and regulatory evasion was concentrated within this coordination layer rather than on the trading floor itself.

For legal counsel, this report translates quantitative findings into a tiered **Deposition Roadmap** and **Discovery Priority List**. It identifies high-priority witnesses based on "betweenness centrality" and temporal activation, recommending a bottom-up deposition sequence starting with the "West Power Operational Core" (*e.g.*, Kate Symes, Bill Williams) before advancing to the "Management-to-Desk Bridge" (*e.g.*, John Lavorato, John Arnold) and the "Coordination Layer" (*e.g.*, Jeff Dasovich, Tana Jones). Additionally, the report isolates specific "bridge edges"—communication channels between distinct departments—that intensified up to 74x during the crisis, flagging these documents for immediate review to establish knowledge and intent across organizational silos. This methodology demonstrates that metadata-driven graph analysis can effectively prioritize discovery and establish evidentiary trajectories in complex corporate litigation.

---

[1] consult@technocrat.site

Historical Background

In 1996, California set out to restructure utility electricity generation in Assembly Bill 1890 (AB 1890). Investor owned electric utilities (primarily PG&E, Southern California Edison Company and San Diego Gas and Electric Company) were directed to divest their generation plan and to purchase power in a newly established Power Exchange (PX) for hourly spot market bidding.

This was, in part, driven by large commercial and industrial customers electing to self-generate, leaving the retail customer base to carry the costs of amortizing existing utility investment in generation, resulting in higher tariff rates. As the proceeds from divestiture were not expected to cover the unamortized value of the plant, utilities would be left with "stranded costs" if unable to raise tariffs to an unacceptable level.

AB 1890 reduced retail rates, freezing them and providing for a special portion of the tariff to recover the stranded costs over several years. The tariff was constituted as a special type of property that could be sold by the utilities in support of debt obligations. This allowed the utilities to issue 10-year bonds at a substantial discount from other available financing.

As an administrative law judge for the California Public Utilities Commission, I was assigned to hear the applications of the three major utilities for authority to issue bonds. My opinion approving the applications was adopted by the Commission.

From discussions with staff concerning the PX implementation, in which I was not involved, the view was that then PX market structure would be "Pareto optimal," meaning, in effect that no market participant could game the system.

Enron negated that view by successfully manipulating the market to its own short-term advantage, but leading directly to its bankruptcy. The Federal Energy Regulatory Commission began to investigate Enron's role in unexpected price increases, and accounting irregularities in the holding company also came to light, leading to securities fraud prosecution by the Securities and Exchange Commission.

As a result of the investigation a collection of approximately 500,000 emails from 146 Enron document custodians was made public, known as the "Enron corpus" (a collection of text documents). I became interested in using the material as a means of applying data science to civil litigation and criminal prosecution.

After reducing the email through deduplication, filtering of spam, normalization of email addresses and automated detection of email chains, I produced smaller set of 249,068 emails. Natural language processing identified a large portion of these as irrelevant (internal expense report approvals, for example). Many more were

broadcast to large lists of receivers. Further restricting selection to internal correspondence between pairs of users who exchanged correspondence, I arrived at a reduced data set.[2] Natural language processing provided unsatisfactory results, which led to a social network analysis that identified potentially useful methods of at least prioritizing the order of discovery. The balance of the text was prepared by ChatGPT Opus 4.6 based on the data file provided. I have reviewed, but not edited it. This is a dramatic demonstration of the capability of artificial intelligence agents to assist in discovery. Exploiting that potential was made possible by human direction, in this case that of the former lawyer and current data scientist who prepared the prompt and provided the data.

This paper is the work of a single evening.

Introduction

This report documents a comprehensive social network analysis of the Enron email corpus, focused on identifying structural and linguistic signatures of market manipulation during the California electricity crisis of 2000–2001. The analysis integrates three complementary methodologies: Louvain community detection on the full internal network, adjacency spectral embedding with Gaussian mixture model clustering (replicating and extending a 2019 latent cluster random effects model), and temporal–lexical analysis of communication patterns across crisis periods.

The Enron corpus comprises approximately 500,000 emails from 142 senior employees. Prior work by the author[3] applied a latent cluster random effects model (ergmm) to a reduced subset of 91 vertices and 1,281 edges, identifying three clusters with distinct vocabularies using only sender/receiver metadata. The present analysis extends this to the full internal penpal network of 2,352 nodes and 4,510 edges, using a cleaned SQL database of 30,000 reciprocal exchanges derived from the original corpus.

Data Pipeline

The penpal network was constructed from the scrubbed Enron corpus by identifying all reciprocal email pairs (addresses appearing as both sender and receiver of at least one message to the same counterpart). Edges were filtered to internal-only (@enron.com) communications. The resulting graph contains 6,284 total penpal pairs with 71,855 messages; the internal-only subgraph comprises 2,364 nodes and 4,520 edges, with a largest connected component of 2,352 nodes.

---

[2] Source code reproduced as Appendix A

[3] Attached as Appendix B

Community Detection

Louvain Algorithm Results

The Louvain algorithm detected 20 communities in the internal network with modularity Q = 0.7721. Intra-community edges account for 72% of all connections (3,246 of 4,510), indicating strong organizational clustering. Community sizes range from 322 (Legal/Compliance) to 3 (smallest micro-communities).

Community-to-organization mapping:

Community	Size	Key Members	Function
Legal/Compliance	322	Shackleton, T. Jones, M. Taylor, Sager	Deal documentation, ISDA
Pipeline/Ops	281	Watson, Hayslett, Corman, Scott	Gas pipeline operations
Trading Mgmt	236	Kitchen, Beck, Lavorato, Zipper, Buy	Trading oversight
Gov Affairs/PR	212	Dasovich, Kean, Steffes, Shapiro	Regulatory/ political strategy
Gas Trading Floor	201	Lenhart, Bass, Hoskins	Physical gas trading
Commercial Legal	188	Nemec, Perlingiere, Tycholiz	Commercial deal legal
Trading Desk	142	Arnold, Quigley, Griffith, Shankman	Gas/power position trading
Trading Ops	129	Presto, Davis, Stokley	Trade settlement/ops
East Gas Sched	113	Germany, Parks, Neal	Eastern gas scheduling
Quant/Research	108	V. Kaminski, J. Kaminski, Crenshaw	Quantitative modeling
West Power	97	Symes, Williams, Guzman	Western power trading
General Counsel	80	Cash, Sanders, Derrick	Litigation/ governance
Project Dev Legal	63	Mann	Project development legal
Structured Finance	54	Rogers	Structured transactions

Centrality Analysis

Betweenness centrality identifies network bridges—individuals who route information between otherwise disconnected communities. The top five by betweenness are:

Person	Betweenness	PageRank	Community
Tana Jones	0.1641	0.02466	Legal/Compliance
Vince Kaminski	0.1110	0.01596	Quant/Research
Gerald Nemec	0.0941	0.01489	Commercial Legal
Jeff Dasovich	0.0895	0.02062	Gov Affairs/PR
Sally Beck	0.0829	0.00925	Trading Mgmt

Tana Jones's dominant betweenness (0.1641) reflects her role as the primary coordinator routing deal documentation across Enron's organizational silos. For the manipulation investigation, Dasovich's position is particularly significant: he bridges Government Affairs with Commercial Legal and West Power Trading, the three communities most directly involved in the California crisis response.

Inter-Community Bridges

The highest-volume cross-community edges reveal the organizational fault lines relevant to manipulation coordination:

Community Pair	Edges	Messages	Top Pair
Commercial Legal ↔ Legal/ Compliance	82	611	Nemec↔Young (79)
Trading Mgmt ↔ Legal/Compliance	61	551	Forster↔Taylor (129)
Trading Mgmt ↔ Trading Desk	48	430	Lavorato↔Arnold (110)
Legal/Compliance ↔ Project Dev	43	351	Clair↔Adams (84)
Gas Trading Floor ↔ Trading Desk	44	290	Grigsby↔Rangel (39)
Gov Affairs ↔ Commercial Legal	16	210	Dasovich↔Tycholiz (49)

The Lavorato↔Arnold edge (110 messages) is a high-volume direct channel between trading management and the star gas trader's desk. Arnold's community (Trading Desk, Community 11) is one of the two priority targets for manipulation detection; the other is West Power (Community 9), which includes Bill Williams, named in FERC's findings on California market manipulation.

Latent Space Models

Comparison: ergmm (2019) vs Spectral Embedding (2026)

The 2019 analysis applied a Bayesian latent cluster random effects model (latentnet::ergmm in R) with a Euclidean two-dimensional latent space and G=3 mixture components. MCMC fitting used 10,000 burn-in and 4,000 sample iterations, yielding an intercept of −0.765, BIC of 2464.52, and transitivity of 0.999. That model operated on 91 vertices selected by stress centrality from the top 100 senders and receivers.

The present analysis replaces the Bayesian ergmm with a frequentist adjacency spectral embedding (ASE) pipeline: normalized Laplacian eigendecomposition followed by Gaussian mixture model clustering. Applied to the top 98 nodes by betweenness centrality (the analogous selection criterion), BIC selects G=3 at d=2, confirming the three-cluster structure of the 2019 model.

Parameter	ergmm (2019)	Spectral+GMM (2026)
Vertices	91	98
Selection criterion	Stress centrality (top 100)	Betweenness centrality (top 100)
Dimensionality	d=2, Euclidean	d=2, normalized Laplacian
Clusters	G=3 (specified)	G=3 (BIC-selected)
Fitting method	MCMC Bayesian	MLE (EM algorithm)
Cluster sizes	Not reported by size	64 / 14 / 20
Uncertain membership (<0.8)	Not reported	5 nodes (5.1%)

Spectral Cluster Interpretation (d=2, G=3)

Cluster 0 — Operations Core (64 members). Gas Trading Floor (13), Trading Mgmt (9), Trading Ops (6), Trading Desk (5), General Counsel (5). Top members: Kay Mann, Chris Germany, Matthew Lenhart, Sally Beck, Eric Bass. This is the transactional execution layer—everyone who touches actual deal flow.

Cluster 1 — Pipeline (14 members). All 14 are Louvain Pipeline/Ops community members. Watson, Hayslett, Corman, Geaccone, Lokay. Membership certainty approaches 100%. This is the most coherent single-function cluster in the network, representing Enron's gas pipeline infrastructure operations.

Cluster 2 — Hub/Coordinator (20 members). West Power (4), Gov Affairs (4), Legal (4), Commercial Legal (2), Trading Desk (2). Top members: Dasovich, Shackleton, Tana Jones, Kaminski, Nemec, Kate Symes. This cluster captures boundary-spanning individuals who bridge organizational silos—the coordination layer through which information flows between functional units.

Higher-Dimensional Models

At d=5, BIC selects G=5, and a critical refinement emerges: the West Power Trading group separates as its own cluster of 7 nodes—Kate Symes, Bill Williams, Dana Davis, Monika Causholli, Cara Semperger, Holden Salisbury, and Mark Forney. This cluster, with 100% membership certainty for all members, represents the California trading desk implicated in FERC's market manipulation findings.

At d=3, G=4 (BIC-optimal), this same West Power cluster appears with 8 members. Its stability across dimensionalities and cluster counts confirms it represents a genuine structural feature of the network, not an artifact of model specification.

Vocabulary Distinctiveness

The 2019 paper's central finding was that graph-based preprocessing successfully identified subgroups with distinct vocabularies using only sender/receiver metadata. We replicate and extend this finding using both subject lines (193,799 messages) and lastword body text (51,657 internal messages with 1,285,075 tokens and 50,866 distinct words).

Louvain Community Vocabulary

Each community exhibits a distinctive lexicon that maps directly to its organizational function. Vocabulary uniqueness ranges from 8% to 27% depending on community size and functional specificity.

Community	Msgs	Vocab	% Unique	Signature Terms
Legal/Compliance	9,271	18,564	26.8%	sita, ibj, ndas, custody, westlaw
Gov Affairs/PR	4,528	15,757	23.9%	ferc, iep, sbx, angelides, noncore, legislature
Pipeline/Ops	4,576	14,025	22.9%	pnr, pueblo, pecos, amarillo, remediation
Quant/Research	1,532	7,516	19.9%	mscf, carnegie, mellon, hjm, informs, boltzmann
Trading Mgmt	4,841	14,810	19.1%	fernley, talent, nomlogic
Commercial Legal	4,346	12,103	18.8%	obligor, aos, nogales, calpeak
Gas Trading Floor	4,042	9,500	18.7%	nuke, coal, alt, tmv, curvefetch
Trading Desk	2,765	9,823	15.6%	candlesticks, autohedge, views, distillates, opec
West Power	3,100	7,514	13.5%	midc, checkout, verballed, offpeak, ladwppx
East Gas Sched	2,027	6,612	13.7%	trco, overinjection, aristech, prearranged

Forensic Vocabulary Findings

West Power Trading. The distinctive vocabulary is a precise fingerprint of California/Western power market activity: midc (Mid-Columbia trading hub), ladwppx (LADWP power exchange), uamps (Utah Associated Municipal Power Systems), checkout (596 occurrences, 641x lift—a power scheduling term), verballed (verbal trade confirmation), and offpeak/hlh/llh (load hour classifications). The high-lift shared terms are dominated by broker names: Prebon (2,587x), Amerex (756x), Bloomberg (1,264x).

Gov Affairs/PR. Regulatory/political response vocabulary: iep (Independent Energy Producers), sbx (California Senate Bill X), angelides (Phil Angelides, California Treasurer who investigated Enron), noncore (customer classification in California deregulation), and reregulation.

Hub/Coordinator cluster (spectral). The word manipulation appears 15 times as a term distinctive to this cluster. It is concentrated in the boundary-spanning coordinator layer—not the trading desks—suggesting that discussion of manipulation occurred at the organizational coordination level rather than among traders themselves.

Comparison with 2019 Paper

Metric	ergmm 2019 (body text)	Spectral 2026 (lastword)
Total distinct words	6,733	50,866
Cluster 1 unique vocab	27.71%	42.1% (Ops Core)
Cluster 2 unique vocab	11.21%	19.0% (Pipeline)
Cluster 3 unique vocab	0%	47.7% (Hub/Coordinator)
Messages analyzed	~13,500	51,657

The higher uniqueness percentages reflect the larger network (98 vs 91 top-centrality nodes), use of lastword rather than full body text (reducing quoted-chain echo), and the full internal penpal network rather than a reduced subset.

Temporal Analysis

Period Definitions

Period	Date Range	Messages	Significance
Pre-crisis	Jun 1999 – May 2000	4,669	Baseline before California crisis
Crisis onset	Jun – Dec 2000	10,811	Price spikes, rolling blackouts begin
Peak crisis	Jan – Jun 2001	13,989	FERC investigations, political crisis
Post-crisis	Jul – Nov 2001	15,556	Enron's financial collapse begins
Bankruptcy+	Dec 2001 – Jun 2002	6,284	Bankruptcy, investigations

Individual Activation Trajectories

Monthly average message counts reveal distinct activation patterns for key individuals:

Person	Pre-crisis	Crisis On	Peak	Post-crisis	Bankr+
K. Symes (WestPwr)	0.0	58.9	256.0	21.8	0.0
J. Dasovich (GovAff)	1.5	92.1	140.3	165.8	11.7
B. Williams (WestPwr)	0.0	0.0	13.3	58.6	9.7
J. Arnold (TradDesk)	1.3	28.7	47.8	95.8	6.3
V. Kaminski (Quant)	28.8	71.0	68.8	0.4	0.0
L. Kitchen (TradMgmt)	3.9	6.9	31.0	164.2	208.0
T. Jones (Legal)	54.5	166.7	163.8	58.6	8.3
G. Nemec (CommLegal)	19.7	49.4	103.8	102.0	12.7
S. Kean (GovAff)	4.4	53.9	55.2	11.2	0.0
J. Lavorato (TradMgmt)	4.8	51.1	24.7	62.6	21.7

Kate Symes shows the sharpest activation curve in the dataset: zero pre-crisis to 256 msgs/month at peak crisis, then collapse to near-zero. This is the signature of a node that activated specifically for the California trading operations. Bill Williams (directly named in FERC findings) was similarly dormant pre-crisis, with activity climbing through 2001. Louise Kitchen shows the opposite pattern—flat during the crisis itself, then exploding to 208 msgs/month at bankruptcy—consistent with managing the collapse rather than the manipulation.

Pair Activation During Crisis

Of the 2,480 penpal pairs active during the crisis window (June 2000–June 2001), 1,843 were entirely new—having no prior communication in the preceding year. The highest-volume new pairs reveal operational clusters forming in real time:

New Pair	Msgs	Community	Note
Symes ↔ Metoyer	488	West Power ↔ West Power	Ops backbone
Symes ↔ K. Thompson	442	West Power ↔ West Power	Ops backbone
Symes ↔ Piwetz	201	West Power ↔ West Power	Ops backbone
Symes ↔ Cason	160	West Power ↔ West Power	Ops backbone
Symes ↔ Hundl	111	West Power ↔ West Power	Ops backbone
Dasovich ↔ Denne	138	Gov Affairs ↔ Gov Affairs	PR response
Dasovich ↔ Kaufman	120	Gov Affairs ↔ Gov Affairs	Political strat
Lenhart ↔ Gillette	148	Gas Floor ↔ Gas Floor	Trading ops

The top five new pairs all involve Kate Symes and are all within West Power. These five connections represent the operational backbone of the California trading desk. They did not exist in the email record before June 2000.

Most Intensified Existing Pairs

Among pairs that existed pre-crisis, the most dramatic intensification:

Pair	Pre	Crisis	Ratio	Communities
Steffes ↔ Dasovich	1	106	106x	Gov Affairs (intra)
Farmer ↔ Parker	1	87	87x	Logistics (intra)
Dasovich ↔ Mara	1	75	75x	Gov Affairs (intra)
Arnold ↔ Lavorato	1	61	61x	TradDesk↔TradMgmt
Dasovich ↔ Alamo	3	147	49x	Gov Affairs (intra)
Arnold ↔ Maggi	1	36	36x	TradDesk↔Portland

The Arnold↔Lavorato edge (61x intensification) is the direct channel between the star gas trader and his management. Arnold↔Maggi (36x) bridges Trading Desk to Portland Trading—the location of Enron's California power trading operations.

Inter-Community Flow Changes

Aggregate message flows between communities during the crisis versus pre-crisis reveal which organizational interfaces activated:

Community Pair	Pre	Crisis	Post	Ratio
West Power ↔ West Power	0	2,093	521	NEW
Project Dev Legal (intra)	0	2,193	247	NEW
EnronOnline (intra)	2	431	74	216x
Trading Ops (intra)	2	279	786	140x
Gov Affairs ↔ Comm Legal	1	74	131	74x
Trading Desk ↔ Comm Legal	1	68	36	68x
Trading Desk (intra)	14	822	1,102	59x
Gas Floor ↔ Pipeline	1	49	40	49x
Trading Desk ↔ Portland	1	43	39	43x
Gov Affairs (intra)	74	2,241	1,846	30x
West Power ↔ Legal/Compl	8	137	16	17x
Trading Mgmt ↔ Trading Desk	12	201	182	17x
Gas Floor ↔ Trading Desk	6	91	96	15x

West Power intra-community traffic went from zero to 2,093 messages during the crisis, then dropped to 521 post-crisis. This community literally did not exist as an active communication cluster before the California electricity crisis began. The cross-community flows that surged most—Gov Affairs↔Commercial Legal (74x), Trading Desk↔Commercial Legal (68x), Trading Desk↔Portland Trading (43x)—are precisely the channels expected if trading strategies required legal coordination and the Portland desk (where California trading was executed) needed to coordinate with Houston.

Vocabulary Shift During Crisis

Lexical analysis of the lastword field reveals the semantic content that filled these newly activated channels.

West Power. Pre-crisis vocabulary comprised only 125 tokens; during the crisis, 77,101. The community's entire lexicon was essentially created during the crisis. Dominant new terms: deal (2,536), peak (681), checkout (594), broker (490), price (444), missing (525). All are operational power trading terms.

Gov Affairs/PR. New terms introduced during the crisis: ferc (236), utilities (157), texas (126), plants (114), enronxgate (112—the internal term for the scandal). California surged 4.6x; electricity surged 8.1x. The term enronxgate later appeared in Trading Desk (200 occurrences), showing the scandal awareness spreading from the political response team to the trading floor.

Trading Desk. New terms: enronxgate (200), shankman (94), mcconnell (75). The name arnold surged 6.0x and john 6.8x, reflecting John Arnold's rising centrality as Enron's star trader during the period.

Use as deposition roadmap[4]

Tier 1: The West Power Operational Core

Kate Symes is the first deposition. She went from zero internal communication to 256 messages per month at peak crisis, then back to zero. She is the hub of five high-volume pairs (Metoyer, Thompson, Piwetz, Cason, Hundl) that collectively didn't exist before June 2000. Her vocabulary is saturated with operational trading terms — checkout, peak, deal, broker, price, missing — and her spectral cluster separates cleanly at every dimensionality tested. She knows the operational mechanics of whatever was happening on that desk.

Bill Williams comes next. Named volume is lower but his activation pattern — dormant pre-crisis, rising steadily through 2001, peaking post-crisis at 58.6 msgs/month — suggests he became more important as the crisis deepened and legal exposure grew. His late-stage activation is consistent with someone who was involved in execution and then drawn into response/cleanup. Depose him after Symes so you can use her testimony to frame questions.

The five Symes counterparties (Metoyer, Thompson, Piwetz, Cason, Hundl) should be deposed as a group, ideally on a compressed schedule. They're the operational layer — scheduling, confirming, executing. Each has near-100% membership certainty in the West Power spectral cluster. They'll have granular knowledge of specific trades and whether standard practices were followed.

Tier 2: The Management-to-Desk Bridge

John Lavorato is critical because of the Arnold↔Lavorato edge (61x intensification, 110 total messages). He sits in Trading Management and connects directly to the Trading Desk. The structural question his deposition answers is: what did management know about trading desk strategies, and when? His trajectory shows two peaks — during crisis onset and again post-crisis — suggesting involvement in both strategy formation and damage control.

---

[4] The agent was instructed to ignore any knowledge in its model concerning the actual prosecutions

John Arnold connects to Lavorato (61x), to Maggi in Portland (36x), and his name surges 6.0x in Trading Desk vocabulary during the crisis. He's the star trader, and the Arnold↔Maggi edge is the bridge between Houston and the Portland desk where California trading was executed. The deposition should focus on the content of those two bridge relationships.

Louise Kitchen shows almost no activity during the crisis itself but explodes to 208 msgs/month at bankruptcy. She was managing the aftermath. Depose her to establish what Trading Management learned and when — her late activation suggests she was brought in once the exposure became apparent, making her a potential witness to admissions or damage-control discussions.

Tier 3: The Coordination Layer

Jeff Dasovich is the most active node in the political response. His pair activations — Steffes (106x), Mara (75x), Alamo (49x), Denne (138 new messages), Kaufman (120 new) — document the real-time formation of the crisis management team. His vocabulary shift to ferc, electricity, california, and the internal coinage enronxgate tells you he was the person synthesizing the political and regulatory exposure. Depose him not about trading mechanics but about what the company understood its exposure to be, when, and what strategy it pursued in response.

Tana Jones has the highest betweenness centrality in the entire network (0.1641) and connects Legal/Compliance to virtually every other community. She routed deal documentation. She can establish the paper trail — what was documented, what required legal review, what was flagged. Her trajectory (54.5 baseline rising to 166.7 during crisis onset) suggests the deal flow was already heavy and intensified rather than appearing from nowhere.

Gerald Nemec bridges Commercial Legal to Legal/Compliance (82 edges, 611 messages with the Young pair alone at 79). His 103.8 msgs/month at peak crisis and vocabulary dominated by obligor, deal-specific entity names, and counterparty terms means he was reviewing the legal structure of the transactions. He can testify to what was unusual or nonstandard about the California-related deals.

Tier 4: Strategic Witnesses

Vince Kaminski (Quant/Research, betweenness 0.111) is valuable because his community has 19.9% unique vocabulary including quantitative modeling terms (hjm, boltzmann, mscf). He drops to 0.4 msgs/month post-crisis — he was pushed out or sidelined. If the quant team flagged risk or raised concerns about trading strategies, Kaminski would know. His disappearance from the network is itself a signal.

Steven Kean (Gov Affairs, 53.9 during crisis onset, drops to 0 at bankruptcy) is senior enough to have had visibility into strategic decisions but his early departure from active communication suggests he was either moved or chose to disengage. Depose late, with the benefit of testimony from Dasovich and the West Power witnesses.

Deposition Sequencing Strategy

The logic is bottom-up: operational witnesses first (Symes, Williams, the Metoyer/Thompson group), then the bridge witnesses who connect operations to management (Lavorato, Arnold), then the coordination/legal layer (Dasovich, Jones, Nemec), then senior strategic witnesses (Kitchen, Kaminski, Kean). Each tier's testimony generates the questions for the next. The network structure tells you which relationships to probe at each deposition — you're essentially walking the graph edges, using each witness to establish what flowed across the connections the data has already identified.

The most valuable single data point for deposition preparation would be the actual message text on the cross-community bridge edges during the crisis window — particularly Lavorato↔Arnold, Arnold↔Maggi, Dasovich↔Tycholiz, and the West Power↔Legal/Compliance edges that surged 17x. Those are the channels where trading strategy met legal review and political management.

Implications for Market Manipulation Detection

Priority Targets

The convergence of structural, lexical, and temporal evidence identifies four priority communities for AI-assisted manipulation detection:

1. West Power Trading (Community 9). The community most directly connected to California market schemes. Activated from zero to 2,093 messages during the crisis. Contains Bill Williams (named in FERC findings) and Kate Symes (highest activation curve in the dataset). Distinctive vocabulary is a precise fingerprint of Western power market operations. Separates as its own latent cluster at d≥3.

2. Trading Desk (Community 11). John Arnold's gas trading desk, with 58.7x intra-community intensification during the crisis. The Arnold↔Lavorato edge (61x) and Arnold↔Maggi edge (36x, cross to Portland) are high-priority channels for textual analysis.

3. Gov Affairs/PR (Community 8). The political and regulatory response network, with 30x intra-community intensification. Dasovich's pair activations with Steffes (106x), Mara (75x), and Alamo (49x) document the real-time formation of the crisis response team. The vocabulary shift to regulatory terminology (ferc, sbx, angelides) confirms the function.

4. Hub/Coordinator spectral cluster. The cross-boundary coordination layer where manipulation appears as a distinctive term. This cluster's 47.7% vocabulary uniqueness and its composition (bridging West Power, Gov Affairs, Legal, and Trading) make it the highest-value target for identifying how knowledge of manipulation strategies propagated across organizational boundaries.

Bridge Edges for Textual Analysis

Cross-community edges that activated or intensified during the crisis represent the channels through which manipulation-relevant information had to flow between organizational silos. The highest-priority edges for AI-assisted textual analysis are:

Trading Desk↔Portland Trading (43x, Arnold↔Maggi), Gov Affairs↔Commercial Legal (74x, Dasovich↔Tycholiz and others), West Power↔Legal/Compliance (17x), Trading Mgmt↔Trading Desk (17x, includes Lavorato↔Arnold), and Gas Trading Floor↔Trading Desk (15x). These edges carry the communication that coordinated trading strategies with legal review and political management.

Methodological Contribution

This analysis validates the 2019 finding that graph-based preprocessing—using only sender/receiver metadata—successfully identifies subgroups with functionally distinct vocabularies. The extension to the full internal network (2,352 vs 91 nodes), combined with temporal analysis, reveals that the community structure is not merely organizational but dynamic: communities activated, intensified, and deactivated in direct response to the California crisis. This temporal signature provides an independent validation that the communities detected are operationally meaningful and directly relevant to market manipulation investigation.

The spectral embedding approach reproduces the ergmm's three-cluster structure at d=2 and, at higher dimensions, reveals additional structure (the West Power cluster) that was obscured by the smaller network in the 2019 analysis. The frequentist approach trades the Bayesian posterior probabilities for computational efficiency, enabling application to the full network—a practical advantage for civil litigation support where rapid iteration is required.

Appendix B: AI-assisted data preparation

```
/*
================================================================================
Enron “penpal” network build-out
Goal
- Start from raw email table `scrub`
- Keep only messages with exactly one “To” and no “Cc” (tosctn=1, ccsctn=0)
- Normalize `tos` from Python-list literal "['email']" into a plain email
field
- Build “penpal pairs”: unordered correspondent pairs (A,B) where both A→B and
B→A exist
- Build “penpal_messages”: all message rows that belong to a penpal pair
- Export messages and graph artifacts (nodes/edges, internal-only, degree,
monthly)

Assumptions
- `sender` is already a plain email address (e.g., swl@winelibrary.com)
- `tos` is stored as ASCII exactly like "['jennifer.stewart@enron.com']"
  so we can strip the first 2 chars and last 2 chars.
================================================================================
*/

/*
--------------------------------------------------------------------------------
1) Create a filtered working table with only one-To / zero-Cc messages.
   This preserves all columns from scrub, but reduces noise/complexity.
--------------------------------------------------------------------------------
*/

DESCRIBE scrub
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| body     | mediumtext   | YES  |     | NULL    |       |
| lastword | mediumtext   | YES  |     | NULL    |       |
| hash     | varchar(250) | YES  | UNI | NULL    |       |
| sender   | varchar(250) | YES  | MUL | NULL    |       |
| tos      | text         | YES  | MUL | NULL    |       |
| mid      | varchar(250) | YES  |     | NULL    |       |
| ccs      | text         | YES  |     | NULL    |       |
| date     | datetime     | YES  |     | NULL    |       |
| subj     | varchar(500) | YES  |     | NULL    |       |
| tosctn   | mediumint    | YES  | MUL | NULL    |       |
| ccsctn   | mediumint    | YES  |     | NULL    |       |
| source   | varchar(250) | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
12 rows in set (0.001 sec)
```

```
DROP TABLE IF EXISTS scrub_1to_0cc;

CREATE TABLE scrub_1to_0cc AS
SELECT *
FROM scrub
WHERE tosctn = 1 AND ccsctn = 0;

/*
-------------------------------------------------------------------------
2) Add parsed recipient column `to_email` and populate it.

   For "['email']":
     SUBSTRING(tos, 3, LENGTH(tos)-4)
   removes leading "['" (2 chars) and trailing "']" (2 chars).
-------------------------------------------------------------------------
*/

ALTER TABLE scrub_1to_0cc
  ADD COLUMN to_email VARCHAR(250);

UPDATE scrub_1to_0cc
SET to_email = LOWER(SUBSTRING(tos, 3, LENGTH(tos) - 4));

/*
-------------------------------------------------------------------------
3) Indexes to speed joins/grouping downstream.
   - idx_sender / idx_to_email: for joins and grouping by endpoints
   - idx_hash: enforce uniqueness for message identity
-------------------------------------------------------------------------
*/

ALTER TABLE scrub_1to_0cc
  ADD INDEX idx_sender (sender),
  ADD INDEX idx_to_email (to_email),
  ADD UNIQUE INDEX idx_hash (hash);

/*
-------------------------------------------------------------------------
4) Create `penpal_pairs` with an explicit schema.

  - CREATE TABLE ... AS SELECT can infer SUM() columns as DECIMAL.
   - Here we want stable types (VARCHAR(250), BIGINT) and a primary key.

   Definition of a “penpal pair”:
   - Unordered pair (p1,p2) where p1 = min(sender,to_email) and p2 = max(...)
   - Must have at least one message in both directions.
-------------------------------------------------------------------------
*/
```



```
DROP TABLE IF EXISTS penpal_pairs;

CREATE TABLE penpal_pairs (
  p1 VARCHAR(250) NOT NULL,
  p2 VARCHAR(250) NOT NULL,
  p1_to_p2 BIGINT UNSIGNED NOT NULL,
  p2_to_p1 BIGINT UNSIGNED NOT NULL,
  total_msgs BIGINT UNSIGNED NOT NULL,
  PRIMARY KEY (p1, p2),
  KEY idx_p1 (p1),
  KEY idx_p2 (p2)
);

/*
-------------------------------------------------------------------------
5) Populate `penpal_pairs` from scrub_1to_0cc.

   - LEAST/GREATEST canonicalize direction into a stable unordered key
   - p1_to_p2 and p2_to_p1 count directional traffic within that pair
   - HAVING clause enforces reciprocity (both directions > 0)
-------------------------------------------------------------------------
*/

INSERT INTO penpal_pairs (p1, p2, p1_to_p2, p2_to_p1, total_msgs)
SELECT
  LEAST(sender, to_email)    AS p1,
  GREATEST(sender, to_email) AS p2,
  SUM(CASE
        WHEN sender = LEAST(sender, to_email)
         AND to_email = GREATEST(sender, to_email)
        THEN 1 ELSE 0 END) AS p1_to_p2,
  SUM(CASE
        WHEN sender = GREATEST(sender, to_email)
         AND to_email = LEAST(sender, to_email)
        THEN 1 ELSE 0 END) AS p2_to_p1,
  COUNT(*) AS total_msgs
FROM scrub_1to_0cc
WHERE sender IS NOT NULL AND sender <> ''
  AND to_email IS NOT NULL AND to_email <> ''
  AND sender <> to_email              -- exclude self-mail
GROUP BY LEAST(sender, to_email), GREATEST(sender, to_email)
HAVING p1_to_p2 > 0 AND p2_to_p1 > 0;

/*

CREATE TABLE penpal_pairs AS
SELECT
  LEAST(sender, to_email)     AS p1,
  GREATEST(sender, to_email)  AS p2,
```



```
  SUM(sender = LEAST(sender, to_email) AND to_email = GREATEST(sender,
to_email)) AS p1_to_p2,
  SUM(sender = GREATEST(sender, to_email) AND to_email = LEAST(sender,
to_email)) AS p2_to_p1,
  COUNT(*) AS total_msgs
FROM scrub_1to_0cc
...
*/

/*
-------------------------------------------------------------------------
6) Create `penpal_messages`:
   All message rows from scrub_1to_0cc whose (sender,to_email) belongs to a
   reciprocal pair in penpal_pairs.Key idea:
   - penpal_pairs is a *summary* table (one row per pair)
   - penpal_messages is the *expanded* message-level table for analysis/export
-------------------------------------------------------------------------
*/

DROP TABLE IF EXISTS penpal_messages;

CREATE TABLE penpal_messages AS
SELECT s.*
FROM scrub_1to_0cc s
JOIN penpal_pairs p
  ON p.p1 = LEAST(s.sender, s.to_email)
 AND p.p2 = GREATEST(s.sender, s.to_email);

/* Indexes for typical analysis (sender/to/date) and message uniqueness */
CREATE INDEX idx_pm_sender ON penpal_messages(sender);
CREATE INDEX idx_pm_to     ON penpal_messages(to_email);
CREATE INDEX idx_pm_date   ON penpal_messages(date);
CREATE UNIQUE INDEX idx_pm_hash ON penpal_messages(hash);

/* Row-count sanity check */
SELECT COUNT(*) AS penpal_message_rows FROM penpal_messages;

/*
-------------------------------------------------------------------------
7) Server-side CSV export (requires secure_file_priv to allow this directory).
   Alternative is client-side export with mysql | sed, shown later.
-------------------------------------------------------------------------
*/

SELECT *
INTO OUTFILE '/opt/homebrew/var/mysql-files/enron.csv'
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
LINES TERMINATED BY ‘\n'
```



```
FROM penpal_messages;

/*
-------------------------------------------------------------------------
8) Client-side CSV export (run in shell, not inside mysql prompt).

mysql -u root -p enron -e "SELECT * FROM penpal_messages" --batch --raw --
skip-column-names \
| sed 's/\t/,/g' > ~/Desktop/enron.csv
-------------------------------------------------------------------------
*/

/*
-------------------------------------------------------------------------
9) Build graph tables:
   - nodes: unique emails involved in penpal_pairs
   - edges: penpal_pairs with numeric node ids (source/target) + weights
-------------------------------------------------------------------------
*/

DROP TABLE IF EXISTS nodes;

CREATE TABLE nodes AS
SELECT DISTINCT p1 AS email FROM penpal_pairs
UNION
SELECT DISTINCT p2 FROM penpal_pairs;

ALTER TABLE nodes
  ADD COLUMN node_id BIGINT UNSIGNED AUTO_INCREMENT PRIMARY KEY,
  ADD UNIQUE KEY idx_email (email);

DROP TABLE IF EXISTS edges;

CREATE TABLE edges AS
SELECT
  n1.node_id AS source,
  n2.node_id AS target,
  p.total_msgs AS weight,
  p.p1_to_p2,
  p.p2_to_p1
FROM penpal_pairs p
JOIN nodes n1 ON n1.email = p.p1
JOIN nodes n2 ON n2.email = p.p2;

ALTER TABLE edges
  ADD INDEX idx_source (source),
  ADD INDEX idx_target (target);
/*
-------------------------------------------------------------------------
```

```
10) Internal-only subgraph
   - nodes_internal: nodes whose email appears in internal.internal
   - edges_internal: edges where both endpoints are internal
-------------------------------------------------------------------------
*/

DROP TABLE IF EXISTS nodes_internal;

CREATE TABLE nodes_internal AS
SELECT n.*
FROM nodes n
JOIN internal i
  ON i.internal = n.email;

ALTER TABLE nodes_internal
  ADD UNIQUE KEY idx_email (email);

DROP TABLE IF EXISTS edges_internal;

CREATE TABLE edges_internal AS
SELECT
  e.source, e.target, e.weight, e.p1_to_p2, e.p2_to_p1
FROM edges e
JOIN nodes_internal ni1 ON ni1.node_id = e.source
JOIN nodes_internal ni2 ON ni2.node_id = e.target;

ALTER TABLE edges_internal
  ADD INDEX idx_source (source),
  ADD INDEX idx_target (target);

/*
-------------------------------------------------------------------------
11) Degree tables
   - node_degree: degree and weighted_degree on full graph
   - node_degree_internal: same metrics on internal-only graph
-------------------------------------------------------------------------
*/

DROP TABLE IF EXISTS node_degree;

CREATE TABLE node_degree AS
SELECT
  n.node_id,
  n.email,
  COALESCE(d.degree, 0) AS degree,
  COALESCE(d.weighted_degree, 0) AS weighted_degree
FROM nodes n
LEFT JOIN (
  SELECT
```

```
    v.node_id,
    COUNT(*) AS degree,
    SUM(v.weight) AS weighted_degree
  FROM (
    SELECT source AS node_id, weight FROM edges
    UNION ALL
    SELECT target AS node_id, weight FROM edges
  ) v
  GROUP BY v.node_id
) d
ON d.node_id = n.node_id;

ALTER TABLE node_degree
  ADD PRIMARY KEY (node_id),
  ADD INDEX idx_degree (degree),
  ADD INDEX idx_wdegree (weighted_degree);

DROP TABLE IF EXISTS node_degree_internal;

CREATE TABLE node_degree_internal AS
SELECT
  n.node_id,
  n.email,
  COALESCE(d.degree, 0) AS degree,
  COALESCE(d.weighted_degree, 0) AS weighted_degree
FROM nodes_internal n
LEFT JOIN (
  SELECT
    v.node_id,
    COUNT(*) AS degree,
    SUM(v.weight) AS weighted_degree
  FROM (
    SELECT source AS node_id, weight FROM edges_internal
    UNION ALL
    SELECT target AS node_id, weight FROM edges_internal
  ) v
  GROUP BY v.node_id
) d
ON d.node_id = n.node_id;

ALTER TABLE node_degree_internal
  ADD PRIMARY KEY (node_id),
  ADD INDEX idx_degree (degree),
  ADD INDEX idx_wdegree (weighted_degree);
```

```
/*
-------------------------------------------------------------------------------
12) Monthly time slicing (dynamic network)
   - edges_monthly: message counts per unordered pair per month
   Uses penpal_messages so the time series is restricted to reciprocal pairs.
-------------------------------------------------------------------------------
*/

ALTER TABLE penpal_messages
  ADD INDEX idx_date (date),
  ADD INDEX idx_sender_to (sender, to_email);

DROP TABLE IF EXISTS edges_monthly;

CREATE TABLE edges_monthly AS
SELECT
  DATE_FORMAT(date, '%Y-%m-01') AS month,
  LEAST(sender, to_email)       AS p1,
  GREATEST(sender, to_email)    AS p2,
  COUNT(*)                      AS msgs
FROM penpal_messages
WHERE date IS NOT NULL
  AND sender IS NOT NULL AND sender <> ''
  AND to_email IS NOT NULL AND to_email <> ''
  AND sender <> to_email
GROUP BY DATE_FORMAT(date, '%Y-%m-01'),
         LEAST(sender, to_email),
         GREATEST(sender, to_email);

ALTER TABLE edges_monthly
  ADD INDEX idx_month (month),
  ADD INDEX idx_pair (p1(250), p2(250));

/*
-------------------------------------------------------------------------------
13) Exports (shell)
   - enron.csv (penpal_messages)
   - enron_edges.csv (penpal_pairs)
   - enron_nodes.csv (nodes)
   - enron_edges_internal.csv, enron_nodes_internal.csv
   - enron_degree.csv
   - enron_edges_monthly.csv
   - enron_edges_monthly_internal.csv (if created)
-------------------------------------------------------------------------------
*/
```

Appendix B: 2019 social network analysis paper

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.

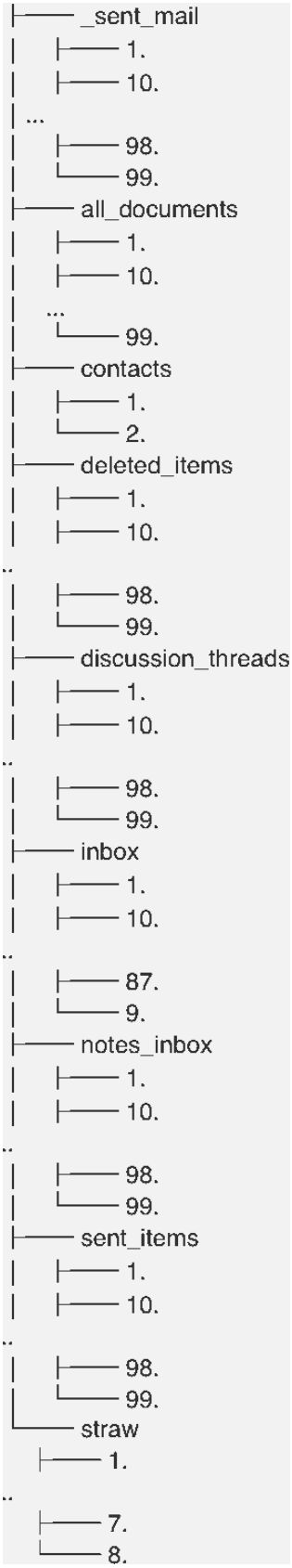


Figure 1: Typical user data

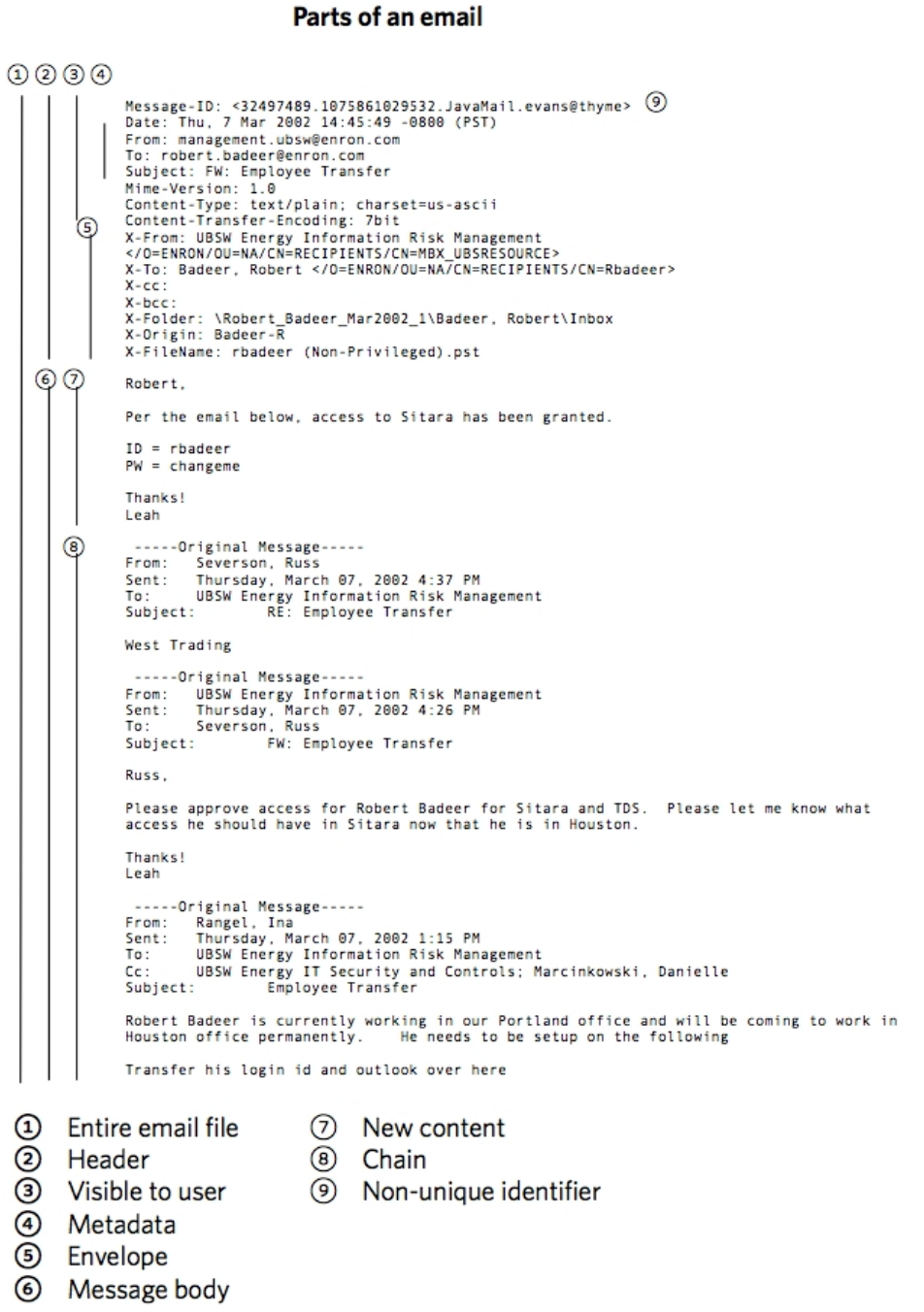


Figure 2: Structural analysis of an Enron email

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.

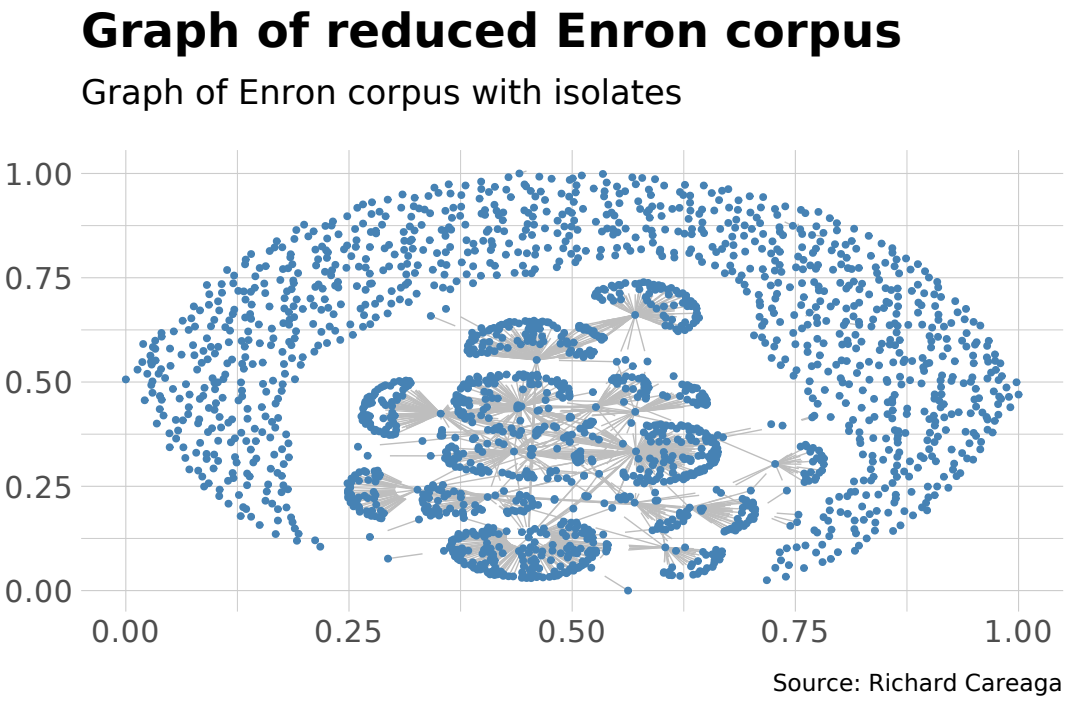
[13] *See, e.g.*, the brief tutorial by Keijo Ruohonen

```
##   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
```

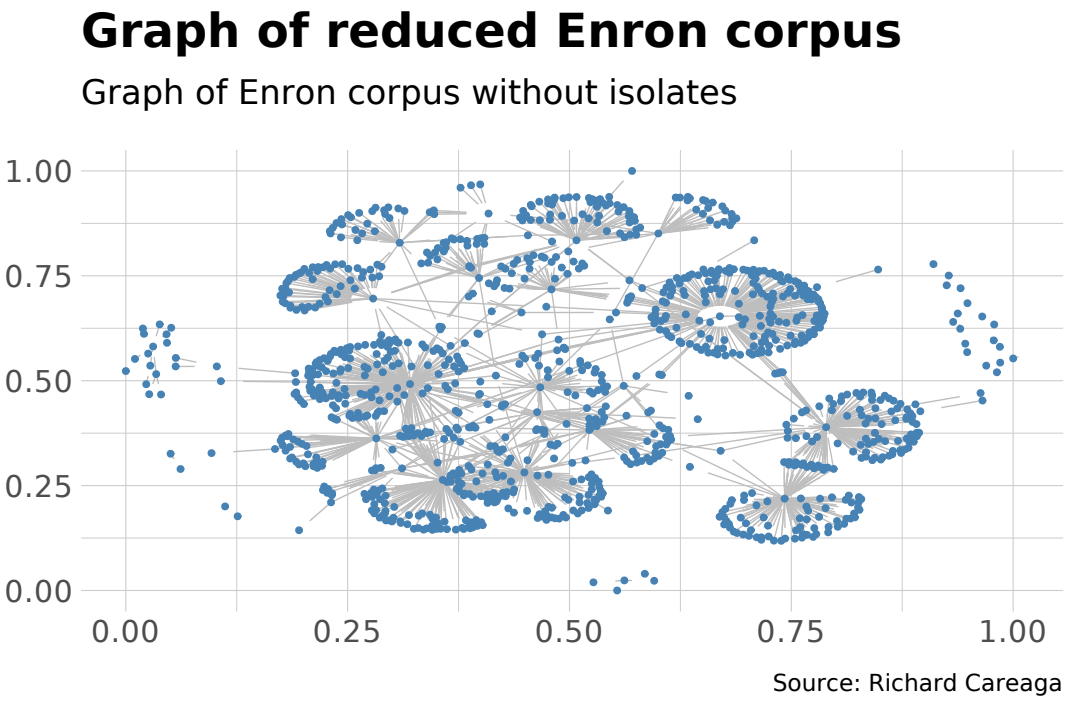
Definition of terms:

- 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.



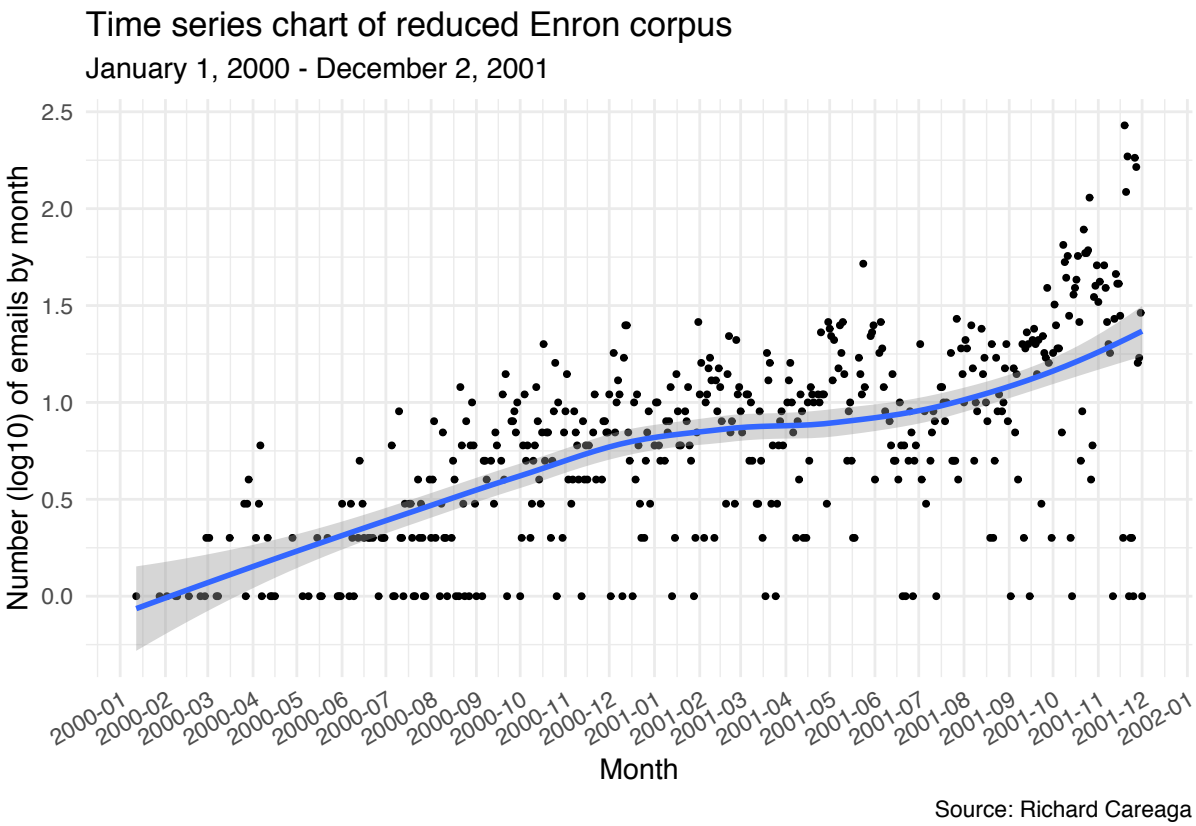
[14]Fruchterman, T. M. and Reingold, E. M. (1991), Graph drawing by force-directed placement. Softw: Pract. Exper., 21: 1129-1164. doi:10.1002/spe.4380211102



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]

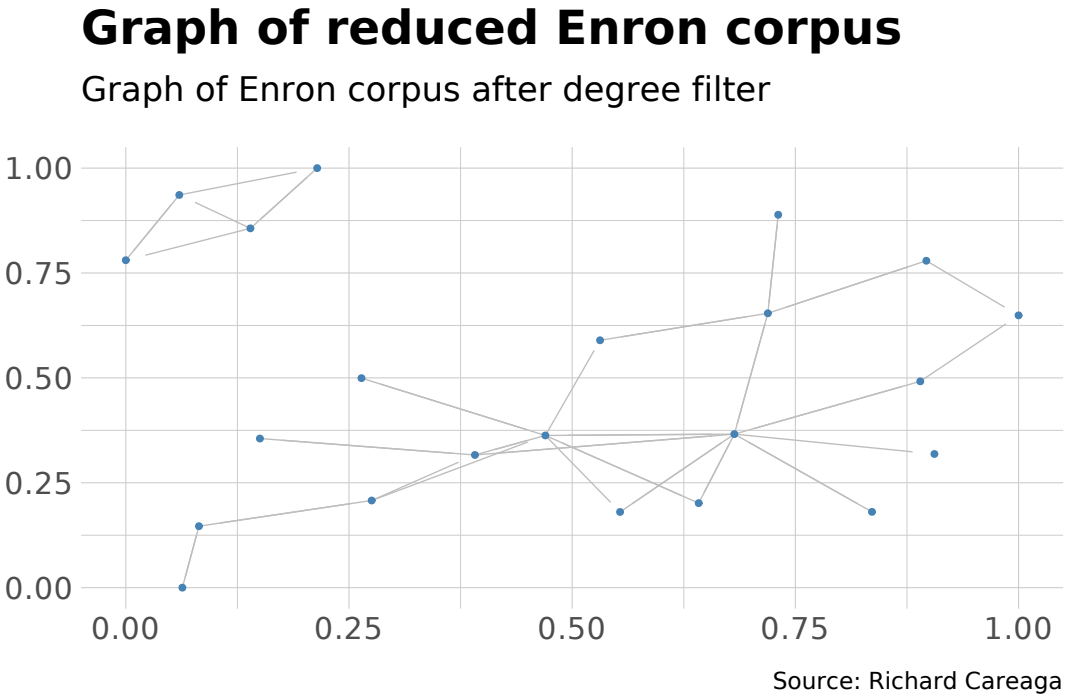
The `sna::stresscent` function is a measure of the shortest number of edges that a vertex has to traverse to reach every other vertex in a graph.

[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.

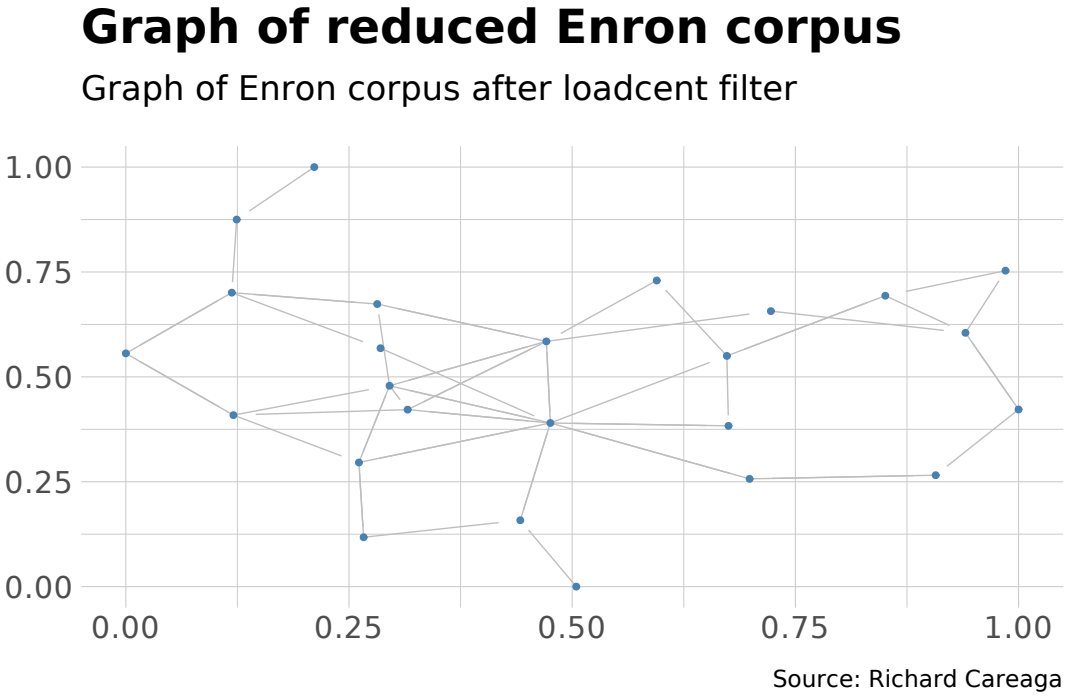
Degree

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



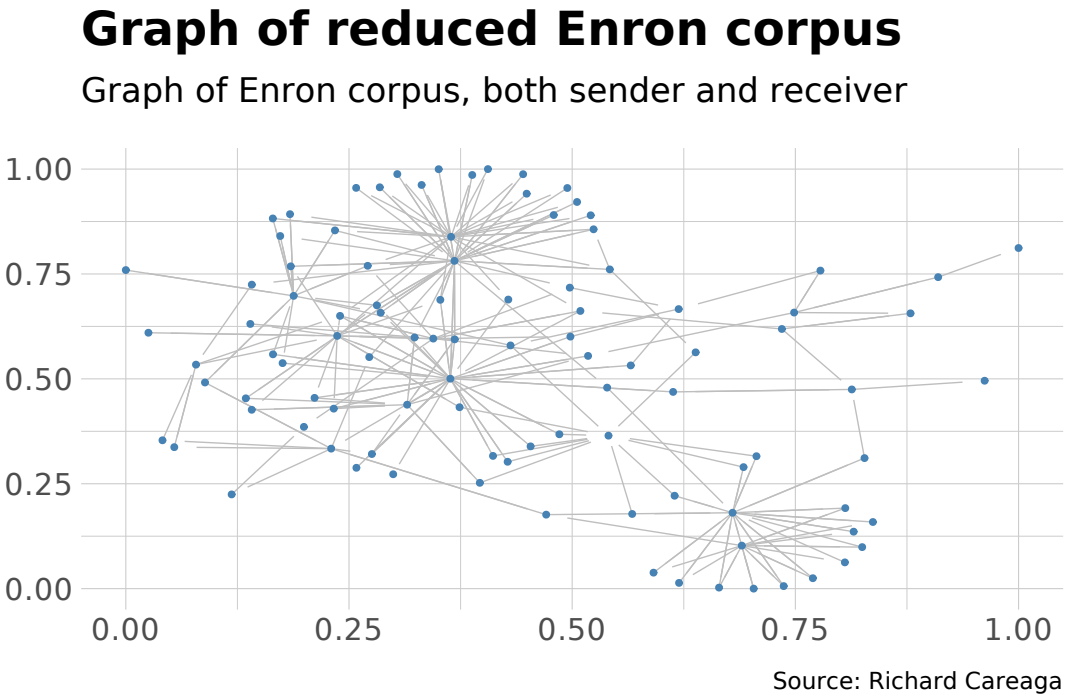
Load centrality

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.

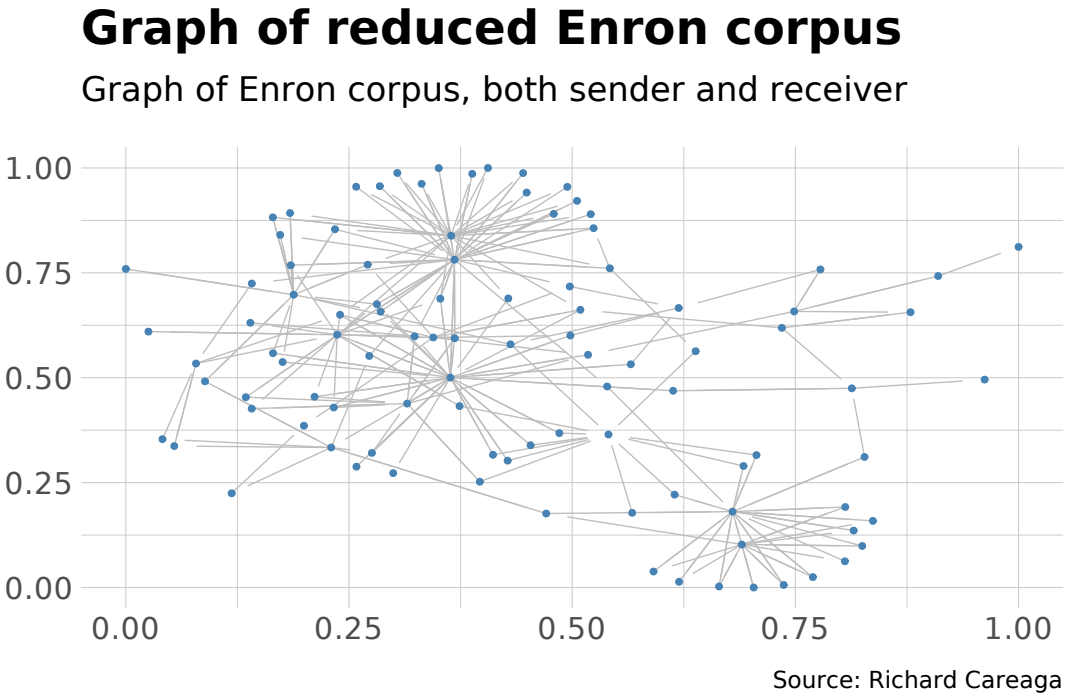
Latent network analysis

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
## ==========================
```

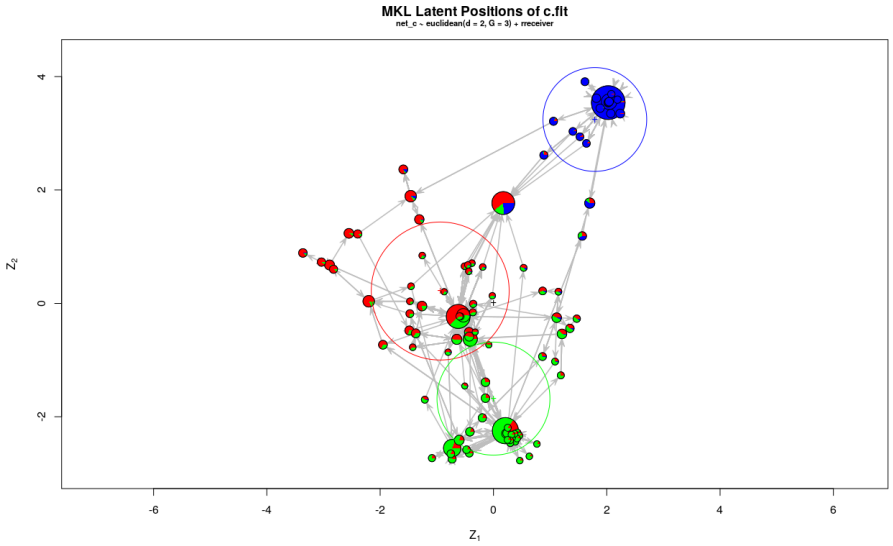


Figure 3: Latent graph based on stresscent

```
##
## 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
```

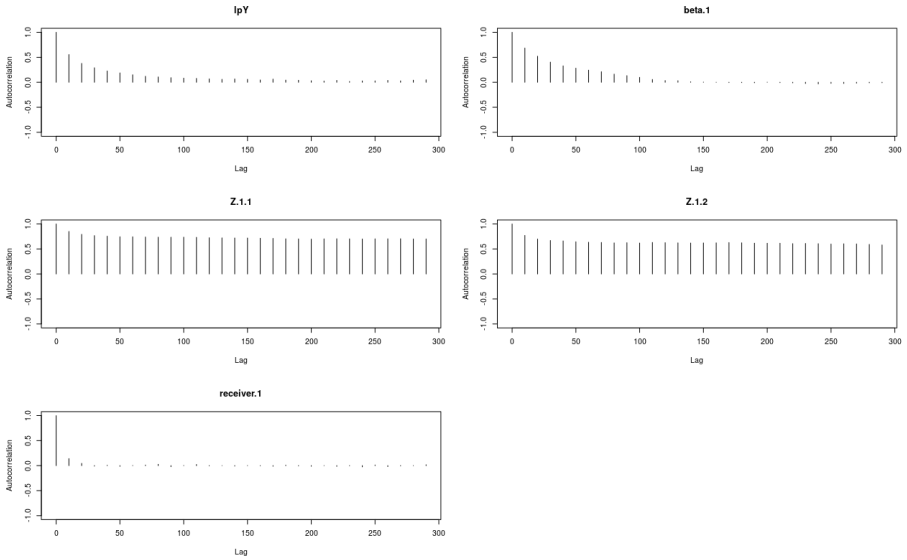


Figure 4: Fit diagnosis part 1

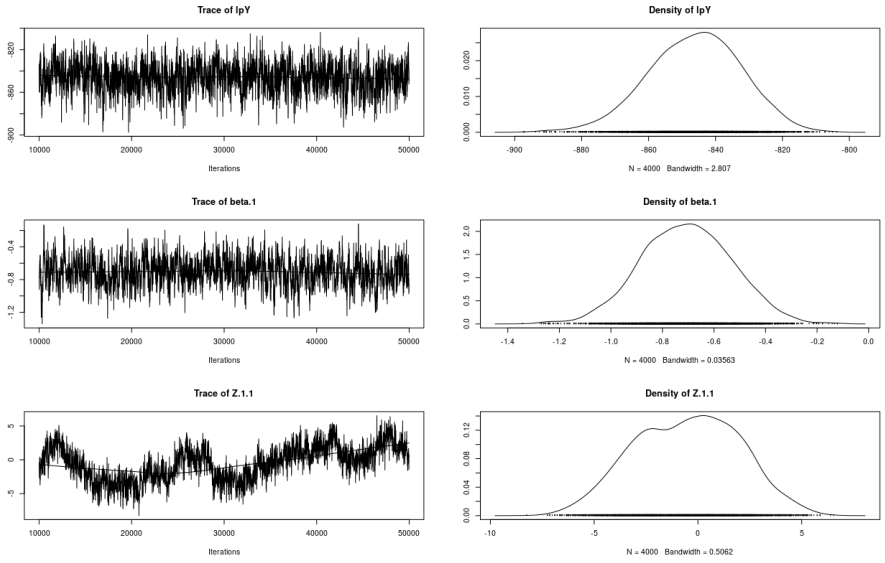


Figure 5: Fit diagnosis part 2

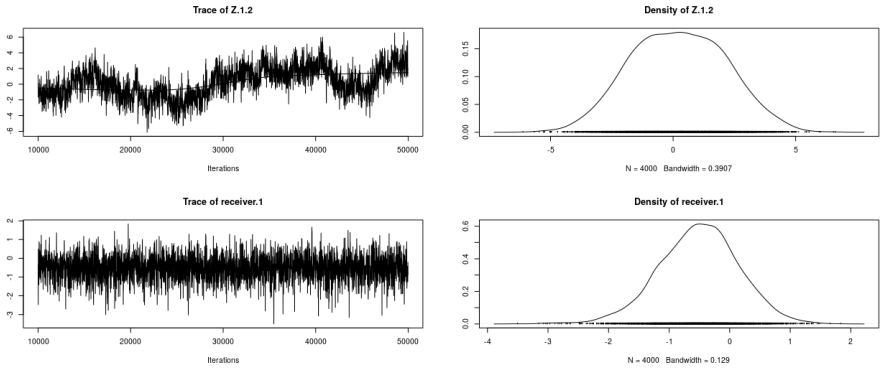


Figure 6: Fit diagnosis part 3

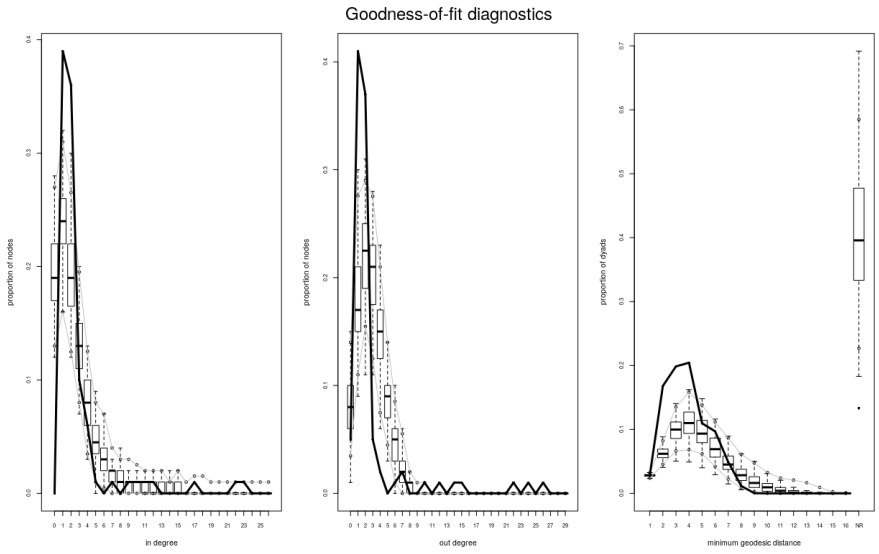


Figure 7: Goodness of fit part 1

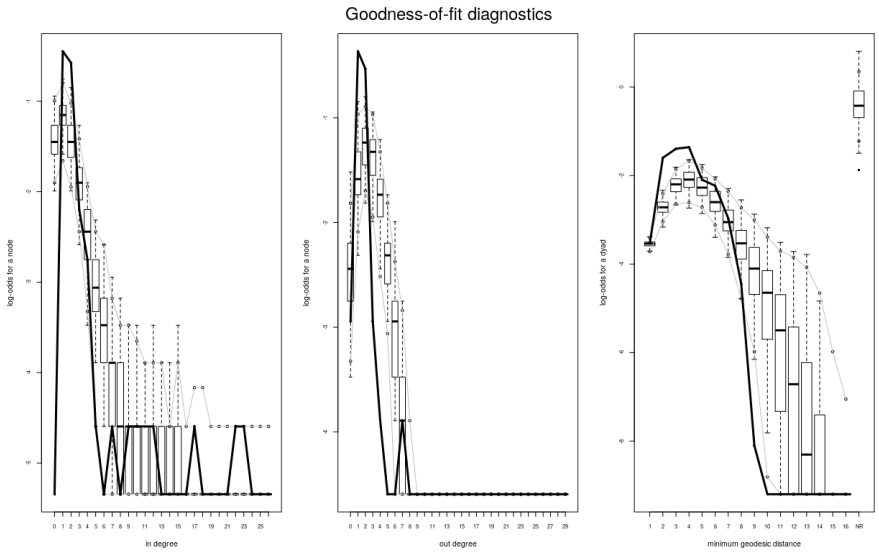


Figure 8: Goodness of fit part 2

```
##
##    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
```

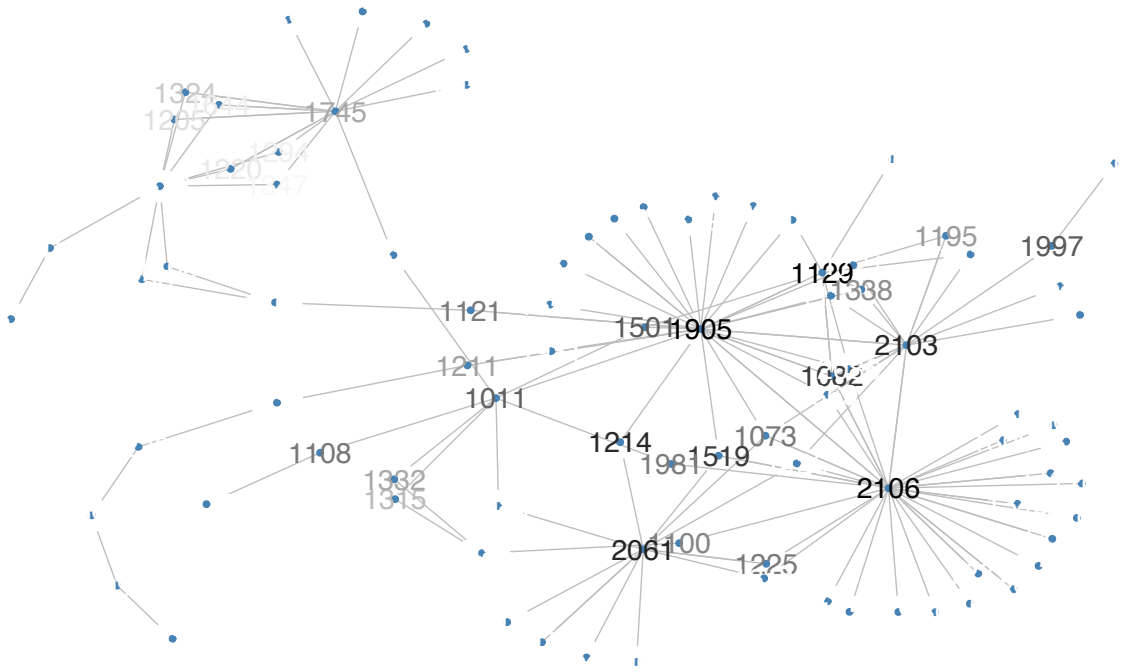
```
## 32   0   0  0.01   1       1.00
## 33   0   0  0.01   1       1.00
##
## Goodness-of-fit for out-degree
##
##    obs min  mean max MC p-value
## 0    5   3  8.32  19       0.28
## 1   41   8 17.25  28       0.00
## 2   37  13 21.77  33       0.00
## 3    5   9 19.48  31       0.00
## 4    2   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

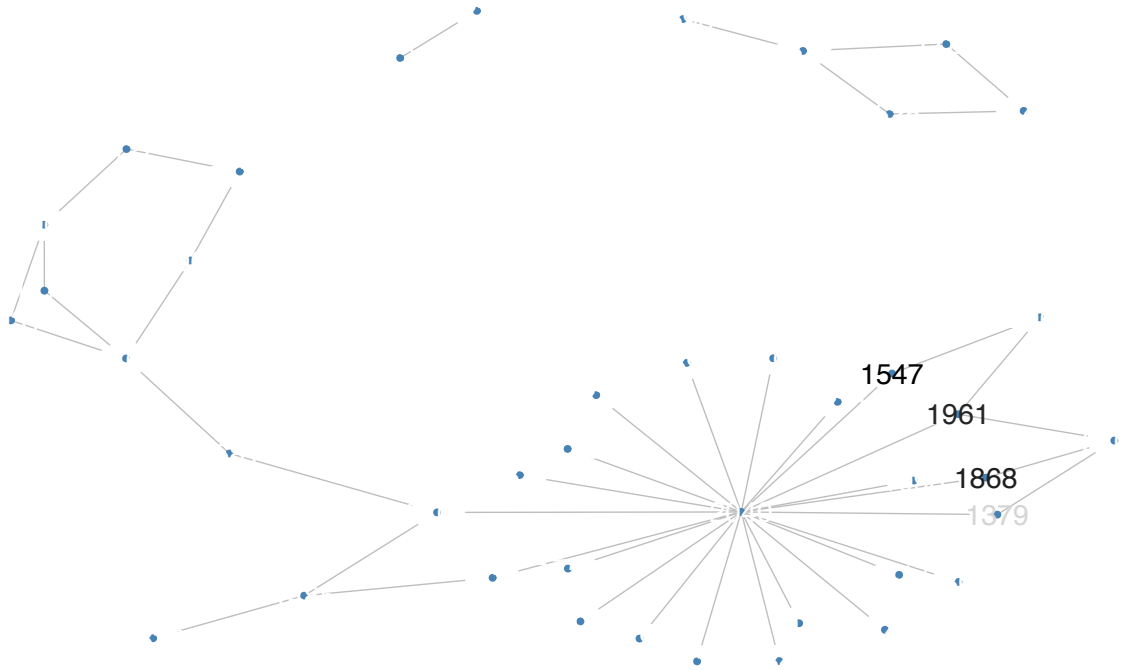
Cluster 1



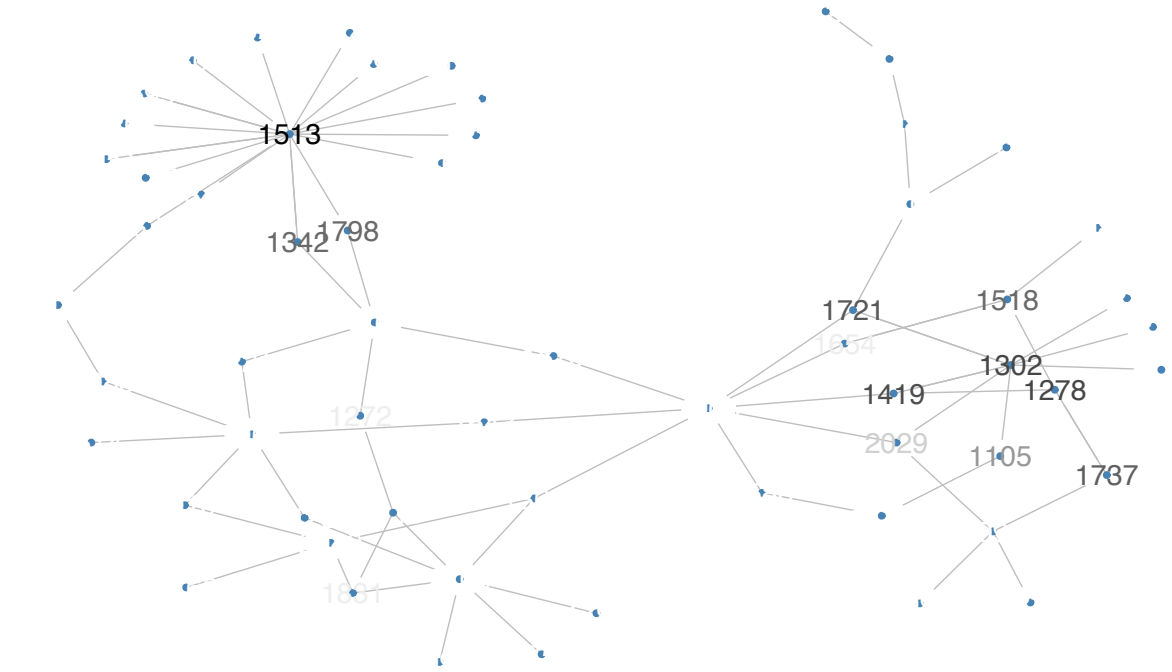
Source: Richard Careaga

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