Text Technologies: Assessment 4
Link Analysis of Enron Emails

Aaron Cronin · s0925570 · The University of Edinburgh

1 Introduction

This report details a system for analysis of 242, 047 emails retrieved from *Enron*'s internal mailing system, following the 2001 scandal.

2 PageRank

2.1 Introduction

PageRank, named after *Google* co-founder Larry Page, is a link-analysis algorithm that estimates how important vertices in a digraph are—given the edges between them. It distributes a flow of *PageRank* over all vertices, such that the total rank sums to 1. It is an iterative algorithm with each node exhausting its rank in order to contribute to its destination vertices, in turn being compensated by vertices linking to it. It eventually converges on a value for all nodes.

2.2 Algorithm

Pseudocode detailing the PageRank implementation used is as follows:

```
for v in V: page_rank[v] = 1 / len(V)
for Iterations:
    l = 0.8
    leaked = l * pr_of_sink_nodes
    rand_pr = 1 - l
    initial = leaked + rand_pr
    for v in V: page_ran'[v] = initial
    for s in Senders:
        R = recipients(s)
        delta = l*page_rank[s]/len(R)
        for r in R:
            page_ran'[r] += delta
    page_rank = page_ran'
```

Dealing with sink nodes In each iteration, the previous iteration's rank for each node is only used to contribute to its children. Therefore the naive version of the algorithm will cause nodes with no children to 'leak' *PageRank*. To counter this, the algorithm above treats all sink nodes as a reservoir of *PageRank* at the beginning of each iteration and then initialises by sharing the combined rank among all vertices on the graph. This, combined with the rank given to every node by the probability of a random visit, makes up the initial rank at each node before nodes linking to it donate a portion of their previous rank.

2.3 Observations

The set of 10 highest *PageRank* score highlights the fact that CEO Kenneth Lay has two active email accounts.

3 Hyperlink-Induced Topic Search

3.1 Introduction

Hyperlink-Induced Topic Search is a link-analysis algorithm that recursively scores vertices using two metrics: hub-score and authority-score.

Hub-score is achieved by linking to many highly-scored authorities. Authority-score is gained from being linked to by many highly-scored hubs. Again, this is iteratively computed until convergence.

3.2 Algorithm

```
initial_score = 1 / sqrt(len(V))
for v in V:
    hub[v] = initial_score
    auth[v] = initial_score
for Iterations:
    for v in V:
        hub[v] = sumauths(outlinks(v))
    normalise(hub)
    for v in V:
        auth[v] = sumhubs(inlinks(v))
    normalise(auth)
```

3.3 Observations

`pete.davis@enron.com` has a hub-score that is far higher than any other. This is due to this account being used as a mailing-list bot that will automatically forward messages to all other *Enron* email accounts.

4 Producing a Graph Visualisation

4.1 Selecting Vertices to Display

The potential vertices, prior to pruning, are the 40 highest *PageRank*ed emails. *PageRank* was chosen as it gives a better indication of who is important in the organisation. Hub-score can be obtained solely by emailing important people, so people who send chain-mail or have at tendency to email everybody will have inflated scores. Auth-score is less exploitable but it doesn't account for the fact that important people tend to engage in bidirectional conversation with each other, so a ranking that takes into account both aspects is preferable.

The decision was made to not merge Kenneth Lay's two email accounts as the visualisation is of messages between accounts and not distinct people.

4.2 Selecting Edges to Display

If all channels of communication between vertices were displayed, the resulting visualisation would be incredibly dense and difficult to interpret. This is the case as even a single message between email accounts would produce a link, and most pairs sent at least one email between themselves over the monitored time-period.

In order to produce a graph with fewer edges, the following heuristic was used:

> Prune any links where the number of emails sent is less than a quarter of a standard deviation above the mean.

This results in only 'significant' communication channels remaining on the visualisation, improving the readibility greatly.

4.3 Pruning Vertices Further

After removing edges that are insignificant, a lot of disconnected vertices appear on the graph. These were pruned by proving `supress_disconnected` as an argument to the `Dot` object. Only 17 of the initial 40 high-influence emails survive this stage.

4.4 Indicating Behaviour of Nodes

Showing Name and Occupation Each node displays the full name and occupation, when possible, of each person by looking them up in the provided employee roster by email-address.

Colour-Transfer of HITS-rank The following transfer function was used to colour each node's record box:

```
auth_score =
    (auth[vl] / highest_auth) * 255.0
hub_score =
    (hub[v] / highest_hub) * 255.0

return to_hex(255.0 - auth_score)
   + '00' + to_hex(255.0 - hub_score)
```

This produces a `RGB` argument to pass to the `color` parameter. The red component is inversely proportional to authority score and the blue component is inversely proportional to the hub score.

Font-size showing PageRank Each node renders the email-owner's details with `fontsize`:

```
5.0 + (5.0 * (pr[e] / mean_pr)))
```

This makes it immediately obvious who the 'key' players are as they have larger nodes on the communication graph.

4.5 Indicating Behaviour of Edges

Pen-width Proportional to Emails Sent After pruning edges with traffic below a quarter-standard-deviation above the mean, edges were drawn proportional to the ratio of traffic to the mean using the `penwidth` parameter.

Arranging Vertices so that High Traffic Links are Short By setting the `weight` parameter to the same value as the `penwidth` parameter, pairs of vertices that have high-traffic links were more likely to be displayed near each other in the resultant rendering.

Phoenix-Rank Named after fictional attorney *Phoenix Wright*, a ranking system was produced that estimates how 'legal-sounding' a collection of documents is. The ranker works by computing a `tf.idf` score for a query consisting of a large wordlist of legal terms. The corpus is the collection of emails extracted from the provided `XML`. Each document to be ranked is the combined tokens of the incoming and outgoing communication between two vertices in the graph.

The following colouring heuristic was used:

> If the combined edges between two nodes have a `phoenix_rank` that is at least a third of a standard deviation above the mean, all directed edges between those nodes are coloured red.

This was effective for finding legal communication channels. It highlighted Mark Taylor as legal-authority and further investigation showed that he was in fact an attorney at *Enron*.

5 Conclusion

The visualisation, provided overleaf, shows clear links between major players in the *Enron* scandal. Most people are predominantly `authorities`, indicated by a pink hue. The exceptions are the two nodes tasked with government affairs and regulatory affairs, and the government relation executive. The affair department's predominant `hub` score shows that they were emailing a lot of people with information and not receiving many replies, perhaps because it was company-wide information broadcast of the scandal as it unravelled. The relation executive is a key player and has high `hub`, `auth` and `PageRank` scores. Pockets of legal communication occur, indicating that some are tasked primarily with formal proceedings and communicate with similar individuals. It is likely that any vertices with all edges coloured red, such as Mark Taylor, has legal training.

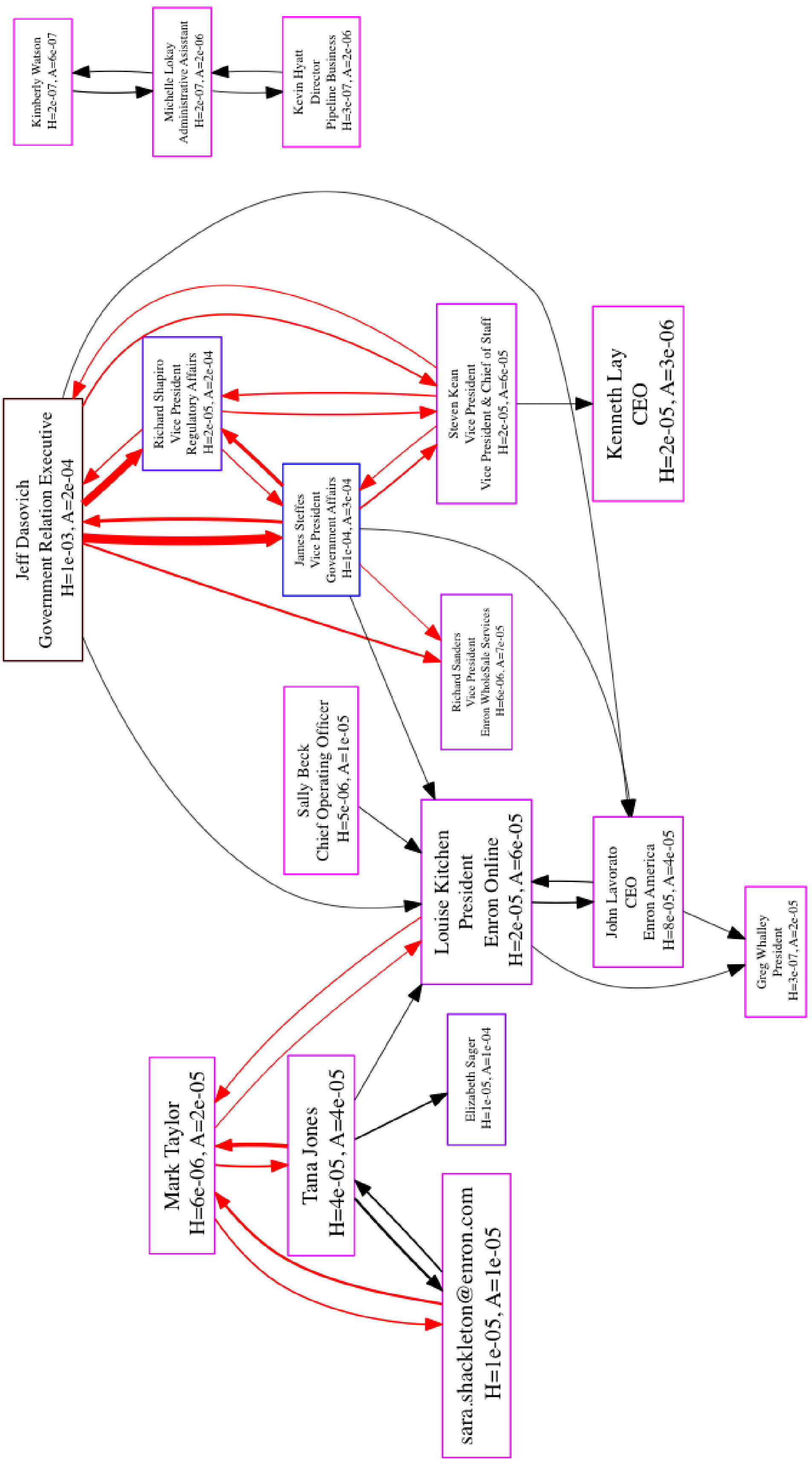
6 Further Work

The `phoenix_rank` edge colouring could be replaced by a system that uses *genre-classification* to label channels with tags such as *social*, *legal*, *formal*, in order to better understand the relationships between employees.

An animation that shows the change in behaviours over time would be interesting as it may highlight drastic changes at the time of the breaking scandal. This could be achieved by breaking the email data into overlapping *windows* and producing keyframes for each.

References

[1] Legal terms wordlist: `www.enchantedlearning.com/wordlist/legal`


Kimberly Watson
H=2e-07, A=6e-07
Michelle Lokay
Administrative Asisstant
H=2e-07, A=2e-06
Kevin Hyatt
Director
Pipeline Business
H=3e-07, A=2e-06
Jeff Dasovich
Government Relation Executive
H=1e-03, A=2e-04
Richard Shapiro
Vice President
Regulatory Affairs
H=2e-05, A=2e-04
Steven Kean
Vice President
Vice President & Chief of Staff
H=2e-05, A=6e-05
Kenneth Lay
CEO
H=2e-05, A=3e-06
James Steffes
Vice President
Government Affairs
H=1e-04, A=3e-04
Richard Sanders
Vice President
Enron WholeSale Services
H=6e-06, A=7e-05
Sally Beck
Chief Operating Officer
H=5e-06, A=1e-05
Louise Kitchen
President
Enron Online
H=2e-05, A=6e-05
John Lavorato
CEO
Enron America
H=8e-05, A=4e-05
Greg Whalley
President
H=3e-07, A=2e-05
Elizabeth Sager
H=1e-05, A=1e-04
Mark Taylor
H=6e-06, A=2e-05
Tana Jones
H=4e-05, A=4e-05
sara.shackleton@enron.com
H=1e-05, A=1e-05