# BAIN_SNA_Text_Mining

Unai Puelles

12/5/2021

## Purpouse and objetive

In this notebook I am going to collect learned knowledge in the subject "Information search and analysis" from the 3rd year of Software Engineering.
I am going to use Enron email dataset, but I am not going to work with all the data. I am going to analyse mails from 2001 onwards. Why this date? If we search some information about Enron, we can see that on December 3, 2001 they declared their bankruptcy, so I think it is a good portion of data to analyse.

## Prepare enviroment

First of all we have to set the workspace directory where we have all the data files and images. Then, load Enron data.

```
load("enron_data_revised.rda")
```

## Explore the data and extract

Firs of all we need to see some information of the dataset we are going to work with.

Dimensions per object

```
dim(edges)
```

```
## [1] 4308    5
```

```
dim(edges.full)
```

```
## [1] 61673     6
```

```
dim(nodes)
```

```
## [1] 149   3
```

Lets see some sample data

```
head(edges.full)
```

```
##                    sender                receiver type
## 1     mary.hain@enron.com sean.crandall@enron.com   TO
## 2     mary.hain@enron.com mike.swerzbin@enron.com   TO
## 3     mary.hain@enron.com robert.badeer@enron.com   TO
## 4 cooper.richey@enron.com robert.badeer@enron.com   TO
## 5     mary.hain@enron.com     m..forney@enron.com   TO
## 6     mary.hain@enron.com robert.badeer@enron.com   TO
##                                                    subject
## 1 Enron s transmission/power exchange model for discussion
## 2 Enron s transmission/power exchange model for discussion
## 3 Enron s transmission/power exchange model for discussion
## 4                                     Change to EnData
## 5                  ISO To Participate in Super Peak Market
## 6                  ISO To Participate in Super Peak Market
##
## 1
## 2
## 3
## 4
## 5 FYI---------------------- Forwarded by Mary Hain/HOU/ECT on 08/29/2000 01:33 PM ----------
## 6 FYI---------------------- Forwarded by Mary Hain/HOU/ECT on 08/29/2000 01:33 PM ----------
##                  date
## 1 2000-08-17 07:11:00
## 2 2000-08-17 07:11:00
## 3 2000-08-17 07:11:00
## 4 2000-08-23 04:39:00
## 5 2000-08-29 06:28:00
## 6 2000-08-29 06:28:00
```

Lets create communities measure for creating a igraph object and edit it with Gephi. We are going to see how the enterprise is distributed and we can understand better the context.

The next image is all enron data separated by communities (colors) and the width of nodes is calculated on the betweenness measure. This was generated with R and Gephi.

### Extract data

First of all we need to format string date of the dataset to R Date object.

```
edges.full$rDate <- as.Date(edges.full$date)
summary(edges.full$rDate)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "1998-11-13" "2000-12-12" "2001-06-11" "2001-05-08" "2001-10-28" "2002-06-21"
```

Let's extract or working dataset from 2001-01-01 until 2002-06-21

```
edges.full.subset <- edges.full[edges.full$rDate >= as.Date("2001-01-01"),]
dim(edges.full.subset)
```

```
## [1] 45215     7
```

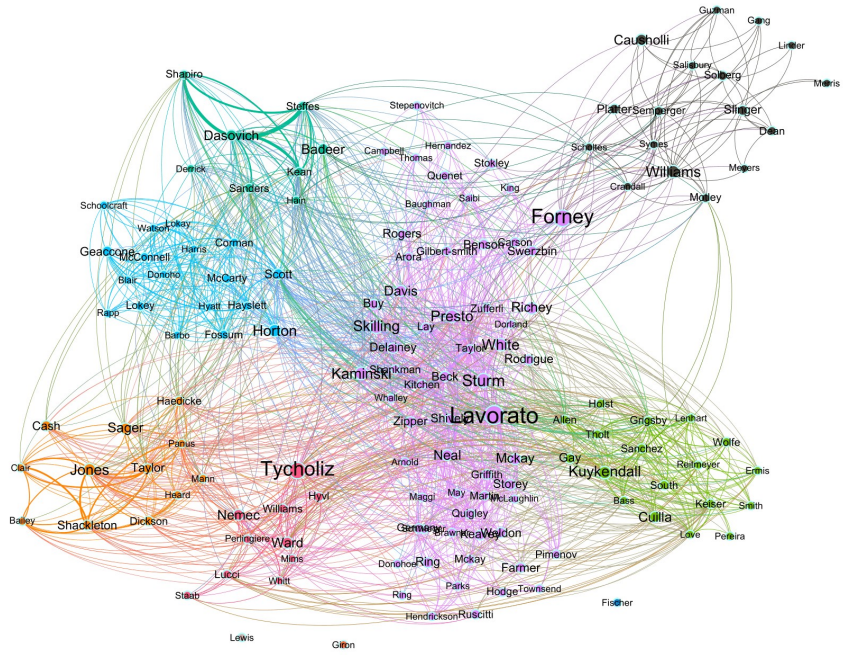



Figure 1: Communities of Enron

```
summary(edges.full.subset$rDate)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2001-01-01" "2001-05-07" "2001-10-01" "2001-08-23" "2001-11-16" "2002-06-21"
```

Let's extract the nodes that appear in the subset we just generated.

```
nodes.subset <- nodes[nodes$Email_id %in% edges.full.subset$sender ||
                        nodes$Email_id %in% edges.full.subset$receiver,]
```

We can see that all the nodes we had appear in the subset we are going to work with.

```
dim(nodes)
```

```
## [1] 149   3
```

# SNA

Now we have our subset generated, let's start doing some SNA.

I am going to create an iGraph object so we can export it to Gephi and create a network image to analyse it.

### Betweenness with iGraph and Gephi

Import necessary iGraph library

```
library(igraph)
```

Create the iGraph object and save it

```
network.subset <- graph.data.frame(edges.full.subset[,c("sender",
                                                        "receiver",
                                                        "type",
                                                        "date",
                                                        "subject")],
                                   directed = TRUE,
                                   vertices = nodes)


write.graph(network.subset,
            file = "enron-subset.graphml",
            format = "graphml")
```

In this image we can see important people in the year of the bankruptcy: Scott, Presto, Grigslay, Taylor, Kitchen...

## SNA Metrics

Now we are going to calculate individual SNA metrics

**Diameter** Is the largest distance between nodes

```
diameter(network.subset)
```

```
## [1] 5
```

**Centrality** We are going to compute Total degree, degree in and degree out.

```
nodes.subset$degree_total <- degree(network.subset,
                                    v = V(network.subset),
                                    mode = c("total"))
nodes.subset$degree_in <- degree(network.subset,
                                 v = V(network.subset),
                                 mode = c("in"))
nodes.subset$degree_out <- degree(network.subset,
                                  v = V(network.subset),
                                  mode = c("out"))
```

Let's see the top 10 of just calculated measures.

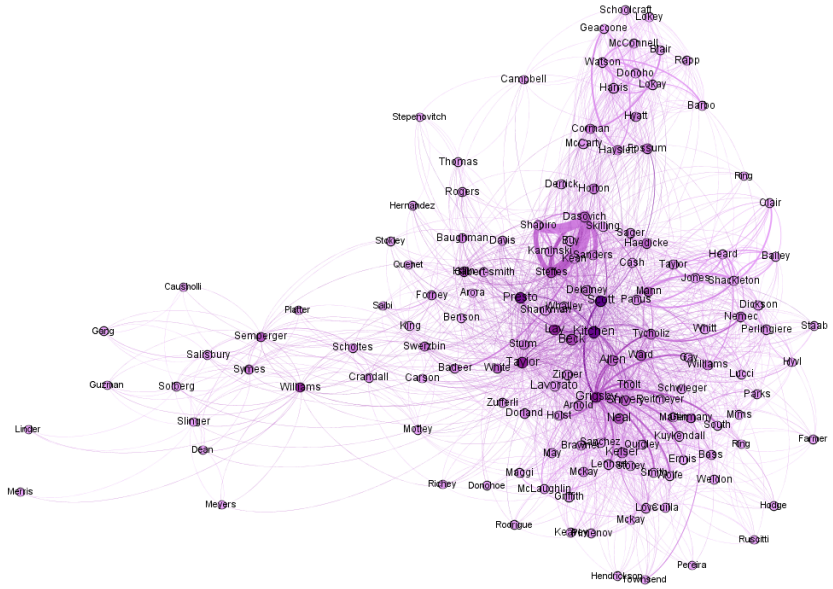



Figure 2: Betweenness centrality of Enron emails from 2001-01 to 2001-06

```
head(nodes.subset[order(nodes.subset$degree_total,
          decreasing = TRUE),], n = 10L)
```

```
##                      Email_id   lastName         status degree_total degree_in
## 42    jeff.dasovich@enron.com   Dasovich       Employee         6906       1084
## 99     mike.grigsby@enron.com    Grigsby        Manager         4563        644
## 36  james.d.steffes@enron.com    Steffes Vice President         4479       2221
## 116 richard.shapiro@enron.com    Shapiro Vice President         3300       2534
## 134   steven.j.kean@enron.com       Kean Vice President         2864       1812
## 14   louise.kitchen@enron.com    Kitchen      President         2724        876
## 125 sara.shackleton@enron.com Shackleton            N/A         2082       1096
## 17  kimberly.watson@enron.com     Watson            N/A         2049        912
## 16       liz.taylor@enron.com     Taylor            N/A         1788         96
## 133 stephanie.panus@enron.com      Panus       Employee         1578        609
##     degree_out
## 42        5822
## 99        3919
## 36        2258
## 116        766
## 134       1052
## 14        1848
## 125        986
## 17        1137
## 16        1692
## 133        969
```

```
head(nodes.subset[order(nodes.subset$degree_in,
                decreasing = TRUE),], n = 10L)
```

```
##                       Email_id   lastName         status degree_total
## 116   richard.shapiro@enron.com    Shapiro Vice President         3300
## 36    james.d.steffes@enron.com    Steffes Vice President         4479
## 134     steven.j.kean@enron.com       Kean Vice President         2864
## 49     barry.tycholiz@enron.com   Tycholiz Vice President         1452
## 125   sara.shackleton@enron.com Shackleton            N/A         2082
## 42      jeff.dasovich@enron.com   Dasovich       Employee         6906
## 135     steven.harris@enron.com     Harris Vice President         1261
## 115 richard.b.sanders@enron.com    Sanders Vice President         1275
## 17    kimberly.watson@enron.com     Watson            N/A         2049
## 14     louise.kitchen@enron.com    Kitchen      President         2724
##     degree_in degree_out
## 116      2534        766
## 36       2221       2258
## 134      1812       1052
## 49       1151        301
## 125      1096        986
## 42       1084       5822
## 135      1032        229
## 115      1027        248
## 17        912       1137
## 14        876       1848
```

```
head(nodes.subset[order(nodes.subset$degree_out,
                decreasing = TRUE),], n = 10L)
```

```
##                       Email_id   lastName         status degree_total degree_in
## 42    jeff.dasovich@enron.com   Dasovich       Employee         6906       1084
## 99     mike.grigsby@enron.com    Grigsby        Manager         4563        644
## 36  james.d.steffes@enron.com    Steffes Vice President         4479       2221
## 14   louise.kitchen@enron.com    Kitchen      President         2724        876
## 16       liz.taylor@enron.com     Taylor            N/A         1788         96
## 17  kimberly.watson@enron.com     Watson            N/A         2049        912
## 134   steven.j.kean@enron.com       Kean Vice President         2864       1812
## 125 sara.shackleton@enron.com Shackleton            N/A         2082       1096
## 133 stephanie.panus@enron.com      Panus       Employee         1578        609
## 123      sally.beck@enron.com       Beck       Employee         1110        182
##     degree_out
## 42        5822
## 99        3919
## 36        2258
## 14        1848
## 16        1692
## 17        1137
## 134       1052
## 125        986
## 133        969
## 123        928
```

Dasovich is an imprtant employee because he has one of the most degrees in and out in this time period.

**Reach 2 step** With this measure we can see the total number of people that person can reach with that number of steps. Here we are going to see the conectivity of each person. Here we can see that the Vice president Presto has mos connections, almost with all the people of the institution.

```
nodes.subset$reach_2_step <-
  neighborhood.size(network.subset,
                    order = 2,
                    nodes = V(network.subset),
                    mode  = c("all"))

head(nodes.subset[order(nodes.subset$reach_2_step,
                decreasing = TRUE),], n = 30L)
```

```
##                        Email_id lastName            status degree_total
## 15      kevin.m.presto@enron.com   Presto    Vice President         1024
## 16          liz.taylor@enron.com   Taylor               N/A         1788
## 26            lavorato@enron.com Lavorato               CEO          374
## 11         kenneth.lay@enron.com      Lay               CEO          554
## 14      louise.kitchen@enron.com  Kitchen         President         2724
## 36     james.d.steffes@enron.com  Steffes    Vice President         4479
## 68    david.w.delainey@enron.com Delainey               CEO          569
## 88         e..haedicke@enron.com Haedicke Managing Director          721
## 110    phillip.k.ellen@enron.com    Allen           Manager         1000
## 117           rick.buy@enron.com      Buy           Manager          342
```

```
## 123         sally.beck@enron.com     Beck          Employee         1110
## 134      steven.j.kean@enron.com     Kean    Vice President         2864
## 49      barry.tycholiz@enron.com Tycholiz    Vice President         1452
## 63          dana.davis@enron.com    Davis    Vice President          254
## 99        mike.grigsby@enron.com  Grigsby           Manager         4563
## 116    richard.shapiro@enron.com  Shapiro    Vice President         3300
## 24           m..forney@enron.com   Forney           Manager          253
## 85         greg.whalley@enron.com  Whalley         President          681
## 80    fletcher.j.sturm@enron.com    Sturm    Vice President          346
## 127         scott.neal@enron.com     Neal    Vice President          590
## 10         keith.holst@enron.com    Holst          Director          600
## 35    hunter.s.shively@enron.com  Shively    Vice President          504
## 48         andy.zipper@enron.com   Zipper    Vice President          370
## 44  jeffrey.a.shankman@enron.com Shankman         President          334
## 132     stanley.horton@enron.com   Horton         President          369
## 139        susan.scott@enron.com    Scott               N/A          996
## 38          jane.tholt@enron.com    Tholt    Vice President          818
## 71        don.baughman@enron.com Baughman           Trader          276
## 75     elizabeth.sager@enron.com    Sager          Employee          662
## 115  richard.b.sanders@enron.com  Sanders    Vice President         1275
##     degree_in degree_out reach_2_step
## 15        341        683          146
## 16         96       1692          145
## 26          3        371          144
## 11        167        387          142
## 14        876       1848          142
## 36       2221       2258          141
## 68        425        144          141
## 88        428        293          141
## 110       694        306          141
## 117       253         89          141
## 123       182        928          141
## 134      1812       1052          141
## 49       1151        301          140
## 63        237         17          140
## 99        644       3919          139
## 116      2534        766          139
## 24         79        174          138
## 85        632         49          138
## 80        233        113          137
## 127       307        283          137
## 10        576         24          136
## 35        398        106          136
## 48        211        159          136
## 44        178        156          135
## 132       196        173          135
## 139       544        452          135
## 38        535        283          134
## 71        109        167          134
## 75        483        179          134
## 115      1027        248          134
```

Now we save them in the network subset object.

```r
nodes.subset$transitivity_ratio <-
  transitivity(network.subset,
               vids = V(network.subset),
               type = "local")

head(nodes.subset[order(nodes.subset$transitivity_ratio,
                  decreasing = FALSE),], n = 20L)
```

```
##                     Email_id lastName         status degree_total degree_in
## 139     susan.scott@enron.com    Scott            N/A          996       544
## 16       liz.taylor@enron.com   Taylor            N/A         1788        96
## 26         lavorato@enron.com Lavorato            CEO          374         3
## 57    chris.germany@enron.com  Germany       Employee          574       100
## 123      sally.beck@enron.com     Beck       Employee         1110       182
## 11      kenneth.lay@enron.com      Lay            CEO          554       167
## 52    bill.williams@enron.com Williams            N/A          379       103
## 14   louise.kitchen@enron.com  Kitchen      President         2724       876
## 7          kim.ward@enron.com     Ward            N/A          988       443
## 22       kam.keiser@enron.com   Keiser       Employee         1028       243
## 23        joe.parks@enron.com    Parks            N/A          162       137
## 56   charles.weldon@enron.com   Weldon            N/A           87        57
## 65   daren.j.farmer@enron.com   Farmer        Manager           75        60
## 42    jeff.dasovich@enron.com Dasovich       Employee         6906      1084
## 15   kevin.m.presto@enron.com   Presto Vice President         1024       341
## 40   jason.williams@enron.com Williams Vice President          833       673
## 84     gerald.nemec@enron.com    Nemec            N/A         1035       514
## 24        m..forney@enron.com   Forney        Manager          253        79
## 62         dan.hyvl@enron.com     Hyvl       Employee          337       133
## 133 stephanie.panus@enron.com    Panus       Employee         1578       609
##     degree_out reach_2_step transitivity_ratio
## 139        452          135          0.1892256
## 16        1692          145          0.2050078
## 26         371          144          0.2128773
## 57         474          128          0.2266667
## 123        928          141          0.2388060
## 11         387          142          0.2675833
## 52         276          112          0.2809524
## 14        1848          142          0.2983051
## 7          545          131          0.3057471
## 22         785          126          0.3167220
## 23          25          124          0.3205128
## 56          30          123          0.3235294
## 65          15          103          0.3333333
## 42        5822          131          0.3478992
## 15         683          146          0.3484043
## 40         160          127          0.3541667
## 84         521          121          0.3600000
## 24         174          138          0.3602941
## 62         204           91          0.3626374
## 133        969          124          0.3666667
```

```
V(network.subset)$outdegree <- degree(network.subset, mode = "out")
V(network.subset)$indegree <- degree(network.subset, mode = "in")
V(network.subset)$degree <- degree(network.subset, mode = "all")
V(network.subset)$reach_2_step <-   neighborhood.size(network.subset,
                                                      order = 2,
                                                      nodes = V(network.subset),
                                                      mode  = c("all"))
V(network.subset)$transitivity_ratio <- transitivity(network.subset,
               vids = V(network.subset),
               type = "local")

V(network.subset)
```

```
## + 149/149 vertices, named, from 5062035:
##   [1] marie.heard@enron.com         mark.e.taylor@enron.com
##   [3] lindy.donoho@enron.com        lisa.gang@enron.com
##   [5] jeff.skilling@enron.com       lynn.blair@enron.com
##   [7] kim.ward@enron.com            kate.symes@enron.com
##   [9] kay.mann@enron.com            keith.holst@enron.com
##  [11] kenneth.lay@enron.com         kevin.hyatt@enron.com
##  [13] joe.quenet@enron.com          louise.kitchen@enron.com
##  [15] kevin.m.presto@enron.com      liz.taylor@enron.com
##  [17] kimberly.watson@enron.com     larry.f.campbell@enron.com
##  [19] larry.may@enron.com           joe.stepenovitch@enron.com
## + ... omitted several vertices
```

# Text Mining

In this enron email data, we aro going to work on the body column that has the content of all the sent emails.

Load necessary Libraries

```
library(quanteda)
library(quanteda.textplots)
library(topicmodels)
library(stringr)
library(quanteda.textstats)
library(ggplot2)
```

Create the corpus for the content of the mails

```
enron.corpus <- corpus(edges.full.subset$body)

head(summary(enron.corpus))
```

```
##    Text Types Tokens Sentences
## 1 text1    73    148         3
## 2 text2    65     88         3
## 3 text3    99    145         7
## 4 text4   144    216         8
```

```
## 5 text5   144    216         8
## 6 text6   167    267        11
```

Now we are going to save the tokens for the matrix creation. We are using some config variables so that we remove words punctuation, nombers and urls for better analysis.

```
words <- tokens(enron.corpus,
                remove_punct = TRUE,
                remove_numbers = TRUE,
                remove_url = TRUE)
```

Now we have to establish the "Stop words". This ones are the ones that will be eliminated from the previous generated words. We are going to use default english provided stopwords and some that we don't want to see. After doing some tests, this ones are the best in my judgement.

```
enron.stopwords <- c(stopwords("en"),
                     as.character(c(0:9)),
                     "<", ">", "=", "$", "+", "s", "na", "t", "d", "also",
                     "subject", "re", "e", "cc", "m", "j", "enron@enron",
                     "ect@ect", "e-mail", "enron", "please", "can", "sent", "message")

words.cleaned <- tokens_remove(words, enron.stopwords)
```

Now, let's generate DFM matrix and see the top features. We can see how gas, power and california are some of the most used words. This three words confirms that Enron was a gas and power provider and they were from California.

```
enron.stemMat <- dfm(words.cleaned)

topfeatures(enron.stemMat, 100)
```

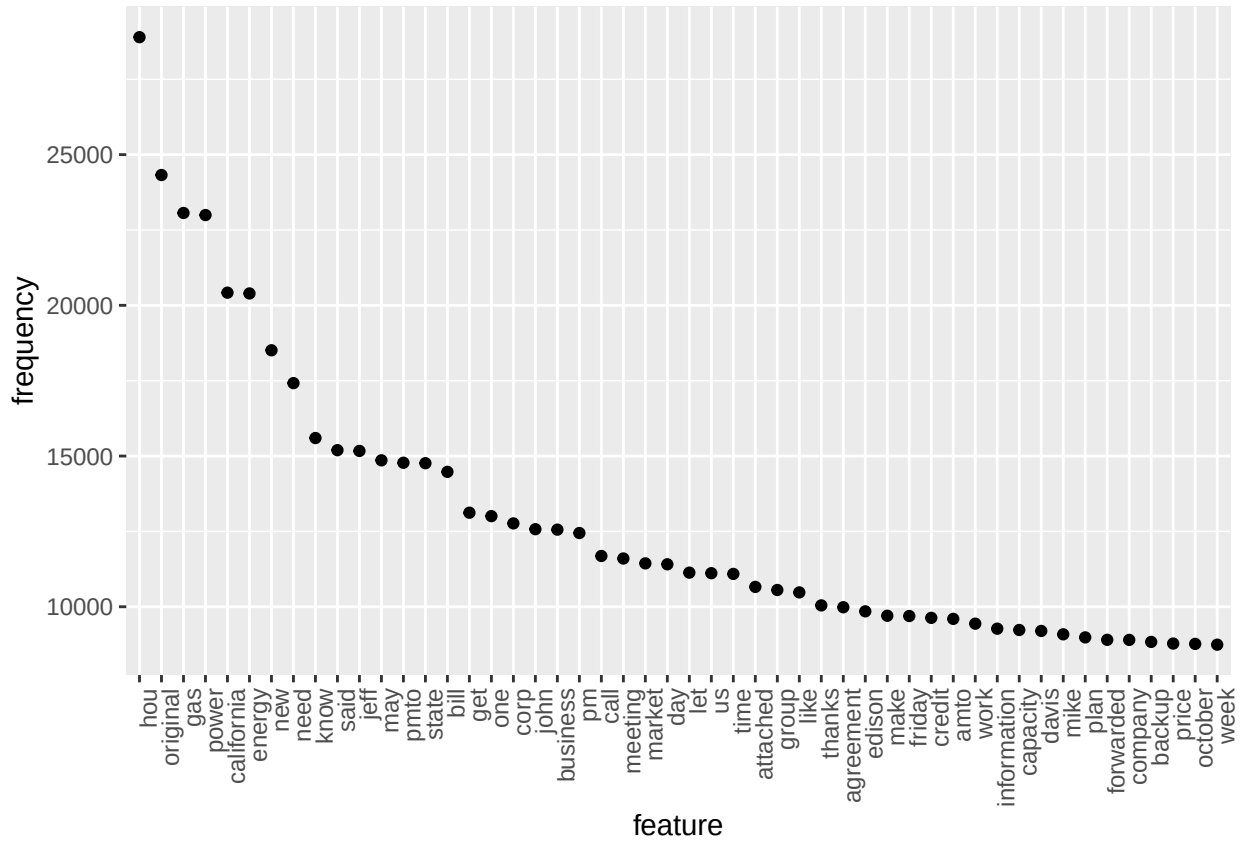
```
## hou original gas
power california
energy
## 28896 24323
23064 22995 20420
20394
## new need know
said jeff may
## 18509 17419
15597 15195 15170
14858
## pmto state bill
get one corp
## 14777 14762
14476 13120 13006
12765
## john business pm
call meeting market
## 12573 12558
12446 11684 11601
11439
```

```
## day let us time
attached group
## 11408 11132
11114 11090 10659
10556
## like thanks
agreement edison
make friday
## 10476 10045 9984
9848 9702 9693
## credit amto work
information
capacity davis
## 9632 9598 9439
9273 9229 9197
## mike plan
forwarded company
backup price
## 9085 8983 8903
8900 8834 8779
## october week
customers rate see
mark
## 8768 8742 8729
8671 8540 8471
## monday dasovich
last today tuesday
contracts
## 8417 8370 8359
8215 8077 7982
## two james
thursday richard
mmbtu back
## 7836 7826 7824
7821 7778 7734
## contract now use
ferc list trading
## 7596 7586 7538
7467 7447 7447
## next wednesday
electricity don
november david
## 7382 7295 7247
7200 7200 7128
## just file
utilities system
think deal
## 7126 7105 7101
7019 6953 6926
## provide
questions access
date available
susan
## 6917 6899 6804
```

```
6731 6725 6719
## tw pg issues
people order want
## 6708 6678 6660
6617 6568 6534
## prices steve
going forward
## 6524 6499 6482
6480
```

Let's show a plot frequency of the words.

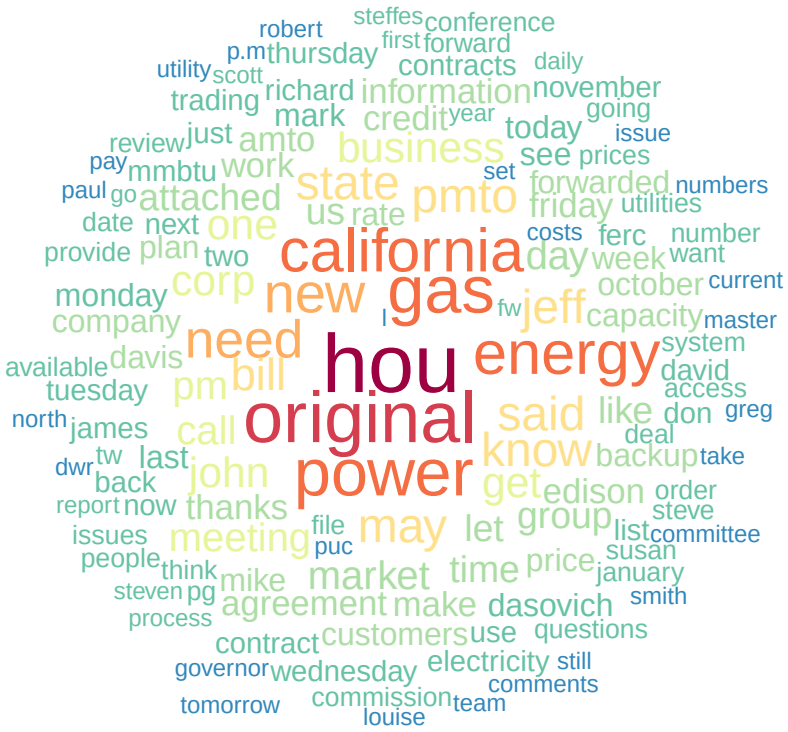
```
enron.stemMat.freq <- textstat_frequency(enron.stemMat, n = 50)

# Sort by reverse frequency order
enron.stemMat.freq$feature <- with(enron.stemMat.freq, reorder(feature, -frequency))

ggplot(enron.stemMat.freq, aes(x = feature, y = frequency)) +
    geom_point() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




Another type of displaying the data is creating a word cloud

```
textplot_wordcloud(enron.stemMat,
                   min_count = 5000,
                   random_order = FALSE,
```

```
                    rotation = 0,
                    color = rev(RColorBrewer::brewer.pal(10, "Spectral")),
                    bg = "black")
```




### Bigrams and trigrams

For more analysis, we are going to do bigrams and trigrams so we can see more context, not like the las visualization that only had one word.

Creating bigram and trigram tokens

```
enron.tokens <- tokens_select(words,
                              pattern = enron.stopwords,
                              selection = "remove")

enron.tokens.ngrams <- tokens_ngrams(enron.tokens,
                              n = 2:3)
```

Now let's create other DFM matrix

```
enron.stemMat.bitrigrams = dfm(enron.tokens.ngrams)
```

Now we are going to do the same visualization for the new matrix

```
print(topfeatures(enron.stemMat.bitrigrams, 100))
```

```
## let_know
## 6843
## jeff_dasovich
## 6550
## north_america
## 3817
## hou_ect
## 3712
## october_pmto
## 3332
## natural_gas
## 3229
## el_paso
## 2833
## james_steffes
## 2701
## direct_access
## 2700
## steffes_james
## 2443
## forwarded_jeff
## 2333
##
forwarded_jeff_dasovich
## 2324
## conference_call
## 2235
## october_amto
## 2159
## phillip_k
## 2157
## smith_street
## 2150
## san_juan
## 2119
## richard_shapiro
## 2064
## grigsby_mike
## 2029
## capacity_mmbtu
## 2029
## backup_location
## 2016
## move_backup
## 2016
## mmbtu_deliveries
## 2007
##
capacity_mmbtu_deliveries
## 2007
## 3d_3d
## 1999
## november_pmto
## 1990
```

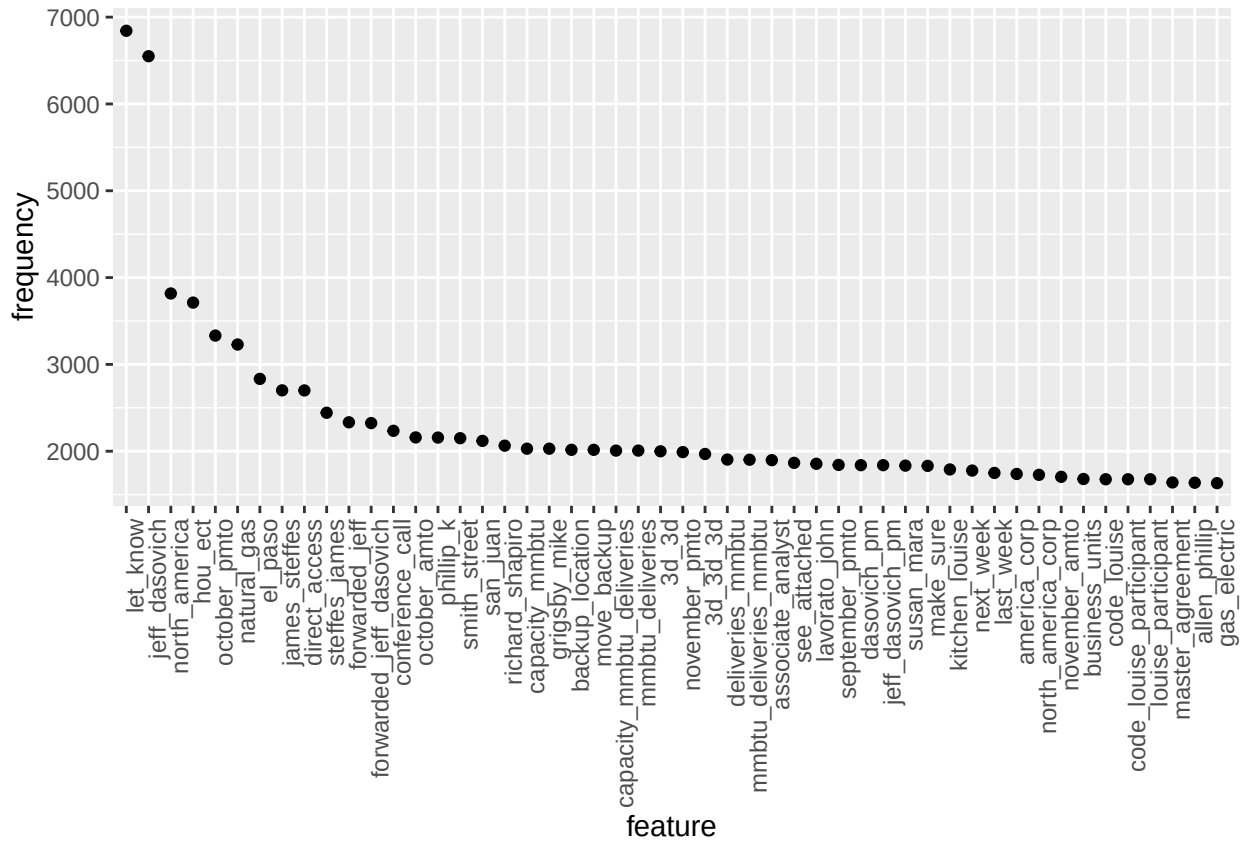
```
## 3d_3d_3d
## 1968
## deliveries_mmbtu
## 1904
##
mmbtu_deliveries_mmbtu
## 1902
##
associate_analyst
## 1897
## see_attached
## 1866
## lavorato_john
## 1855
## september_pmto
## 1842
## dasovich_pm
## 1839
## jeff_dasovich_pm
## 1839
## susan_mara
## 1834
## make_sure
## 1831
## kitchen_louise
## 1790
## next_week
## 1777
## last_week
## 1750
## america_corp
## 1738
##
north_america_corp
## 1728
## november_amto
## 1705
## business_units
## 1680
## code_louise
## 1677
##
louise_participant
## 1677
##
code_louise_participant
## 1677
## master_agreement
## 1640
## allen_phillip
## 1637
## gas_electric
## 1632
## steven_kean
```

```
## 1614
## monday_october
## 1604
## thanks_liz
## 1548
## watson_kimberly
## 1489
## new_york
## 1487
## business_unit
## 1483
## allen_phillip_k
## 1467
## ect_pm
## 1461
## dasovich_jeff
## 1452
##
taylorassistant_greg
## 1440
##
greg_whalley713.853.1935
## 1440
##
whalley713.853.1935_office713.853.1838
## 1440
##
office713.853.1838_fax713.854.3056
## 1440
##
fax713.854.3056_mobile
## 1440
##
taylorassistant_greg_whalley713.853.1935
## 1440
##
greg_whalley713.853.1935_office713.853.1838
## 1440
##
whalley713.853.1935_office713.853.1838_fax713.854.3056
## 1440
##
office713.853.1838_fax713.854.3056_mobile
## 1440
## gas_daily
## 1432
## new_building
## 1417
## p.m_cst
## 1390
## tuesday_october
## 1365
##
deliveries_california
```

```
## 1352
## new_business
## 1351
## 30th_31st
## 1350
## backup_seat
## 1344
## backup_seats
## 1344
## test_times
## 1344
## seat_assignment
## 1344
##
move_backup_location
## 1344
## hou_ect_pm
## 1343
## california_mmbtu
## 1338
## large_pkgs
## 1338
## bill_bradford
## 1333
##
average_deliveries
## 1332
##
deliveries_california_mmbtu
## 1332
##
average_deliveries_california
## 1326
## feel_free
## 1323
## sue_mara
## 1308
## paul_kaufman
## 1302
## street_eb
## 1298
## smith_street_eb
## 1298
## presto_kevin
## 1288
## numbers_listed
## 1277
## dial-in_numbers
## 1268
## soon_possible
## 1266
##
southern_california
## 1266
```

```
## analyst_program
## 1264
##
associate_analyst_program
## 1264
##
dial-in_numbers_listed
## 1263
```

```
enron.stemMat.bitrigrams.freq <- textstat_frequency(enron.stemMat.bitrigrams, n = 50)

# Sort by reverse frequency order
enron.stemMat.bitrigrams.freq$feature <- with(enron.stemMat.bitrigrams.freq, reorder(feature, -frequency

ggplot(enron.stemMat.bitrigrams.freq, aes(x = feature, y = frequency)) +
    geom_point() +
    theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
textplot_wordcloud(enron.stemMat.bitrigrams,
                   min_count = 1500,
                   random_order = FALSE,
                   rotation = 0,
                   color = rev(RColorBrewer::brewer.pal(10, "Spectral")),
                   bg = "black")
```


From this bigrams and trigrams we can extract that Jeff Dasovich was an important pearson. Doing some research on Internet, we can see that Jeff was one of the primary interlocutors inside the company and CEO of Enron.

### Topics generation

Import required libraries

```
library(quanteda.textstats)
library(RColorBrewer)
library(wordcloud)
```

Generate topicmodels. WARNING!! High computation needed (20GB RAM +-)

```
quant_dfm <- dfm_trim(enron.stemMat.bitrigrams,
                      min_termfreq = 10)


if (require(topicmodels)) {
  topicmodels.fit <- LDA(convert(quant_dfm, to = "topicmodels"),
                      k = 4)
  get_terms(topicmodels.fit, 5)
}
```

```
##      Topic 1          Topic 2          Topic 3
```

```
## [1,] "3d_3d"          "let_know"       "backup_location"
## [2,] "3d_3d_3d"       "steffes_james"  "move_backup"
## [3,] "p.m_cst"        "october_pmto"   "capacity_mmbtu"
## [4,] "new_business"   "hou_ect"        "mmbtu_deliveries"
## [5,] "numbers_listed" "kitchen_louise" "capacity_mmbtu_deliveries"
##      Topic 4
## [1,] "jeff_dasovich"
## [2,] "forwarded_jeff_dasovich"
## [3,] "forwarded_jeff"
## [4,] "hou_ect"
## [5,] "dasovich_pm"
```

Let's generate topicmodels wordclouds

```
kk <- topicmodels.fit@beta
# Generamos una matriz de dimensión k (tópicos) = 12 y n tokens (70k)
class(kk)
```

```
## [1] "matrix" "array"
```

```
dim(kk)
```

```
## [1]      4 252447
```

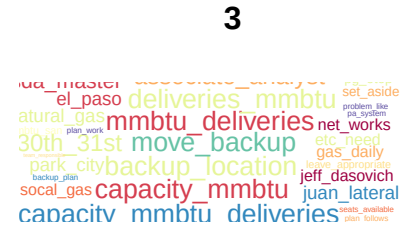
```
# Para poder dibujar los wordclouds ponemos el token como nombre de columna
colnames(kk) <- topicmodels.fit@terms
kk[, 5:10]
```

```
##      friday_march march_amto badeer_robert robert_friday amto_grigsby
## [1,]    -9.522889  -9.828961     -20.42928     -9.987543   -13.956715
## [2,]   -10.019216  -9.961348     -11.08567    -19.828924   -10.091374
## [3,]   -10.591085 -10.567577     -21.66651    -20.048290    -9.181013
## [4,]    -9.324747 -10.420239     -10.67268    -38.759565    -9.975994
##      grigsby_mikesubject
## [1,]          -17.149291
## [2,]           -8.752870
## [3,]           -9.311192
## [4,]          -10.605915
```

```
par(mfrow=c(2, 2))

for (k in 1:length(kk[,1])) {

  topic1 <- kk[k,]

  v <- topic1

  # utilizando rank pasamos el beta numérico a orden (entero, positivo)
  d <- data.frame(word = names(v), rank= rank(v))

  # ordenamos descendente (por defecto -sin el "-" es ascendente)
  d <- d[order(-d$rank),]
```

```
  # normalizamos (parecido a una frecuencia de palabras) +100 para que tenga rango amplio
  d$freq <- d$rank - max(d$rank) + 100

  # Now with a prettier layout
  # baed on code published in
  # http://onertipaday.blogspot.com.es/2011/07/word-cloud-in-r.html
  #plot.new()

  pal2 <- brewer.pal(11,"Spectral")
  wordcloud(d$word,
            d$freq,
            # scale nos da la diferencia relativa (máx mín) entre tamaños de palabras
            scale = c(1, 0.1),
            # max.words las que quepan
            max.words = 100,
            random.order = FALSE,
            rot.per = 0,
            colors = pal2,
            random.color = TRUE)
  title(main = paste(k),
        font = 7,
        col.main = "black")
}
```







We can see 4 topicmodels and their respective wordclouds. The first topicmodel we can say that they are

institutions and names of people. The second topic is the products the can sell. On the other hand, we can see actions like backup and in the fourth one, sender tipoc model (main objetive California).

I hope this project allow you to understand better Enron data.

---

Unai Puelles López