FIT3152 Assignment 3

Name: Ying Qi Mah

Student ID: 32796765

Task 1

I have collected a total of 15 different documents from 3 different topic areas. 5 of the documents are about crime news, 5 are about movie reviews and 5 are short stories. They are all copied-text from web-based articles and saved as text file with the name (type)(number).txt.

Task 2

All of my documents are in txt format, so they can be read directly into R, hence no further process needs to be taken. All the documents are named as their type, followed by a number from 1 to 5 for easy identification. Next, I created the corpus for all 15 documents. (Refer to code at appendix [1])

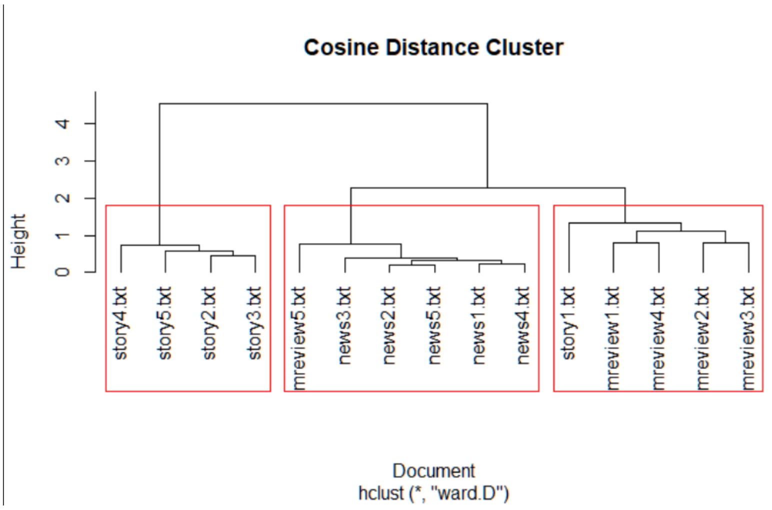
```
 [1] "mreview1.txt" "mreview2.txt" "mreview3.txt" "mreview4.txt" "mreview5.txt" "news1.txt"    "news2.txt"    "news3.txt"    "news4.txt"
[10] "news5.txt"    "story1.txt"   "story2.txt"   "story3.txt"   "story4.txt"   "story5.txt"
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 15
```

Task 3

I have removed the numbers, punctuation, white space in all of the documents. Next, I transformed all the characters in to lower case, removed stop words and stemmed some English words. After doing these, I created a document term matrix. The particular text transformations that I have done is, I found out 30 of the most frequent words from the document term matrix and removed them to have a better performance in clustering. Then I have remove some sparse terms. Finally, after removing sparse terms, I am left with 33 terms to work with. The reason I chose 33 terms is because it gives to best result in clustering. And it is also due to removing sparse terms gives me only the choice of 12 terms and 33 terms.(Refer to appendix[2] for the code) (Refer to DTM table at the end of the document)

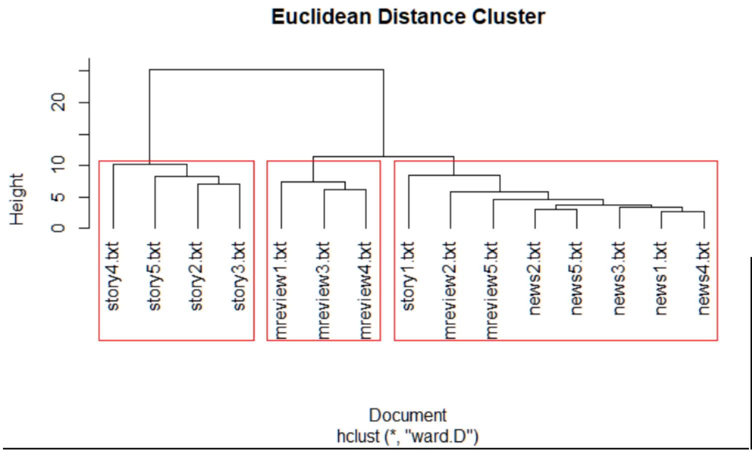
Task 4

I have done both Cosine Distance clustering (Refer to code at appendix[3]) and Euclidean Distance clustering (Refer to code at appendix[4]) to compare and contrast.



	Clusters		
GroupNames	1	2	3
News	0	5	0
Reviews	4	1	0
Stories	1	0	4

I have also constructed the confusion matrix for the result. From the confusion matrix, we can calculate the accuracy, (5+4+4)/15 = 0.8666. Hence, the accuracy of cosine distance cluster is 0.866



	Clusters		
GroupNames	1	2	3
News	0	5	0
Reviews	3	2	0
Stories	0	1	4

For Euclidean Distance Cluster, the accuracy is (5+3+4)/15 = 0.8. As a result, the accuracy of the Euclidean Distance Cluster is 0.8

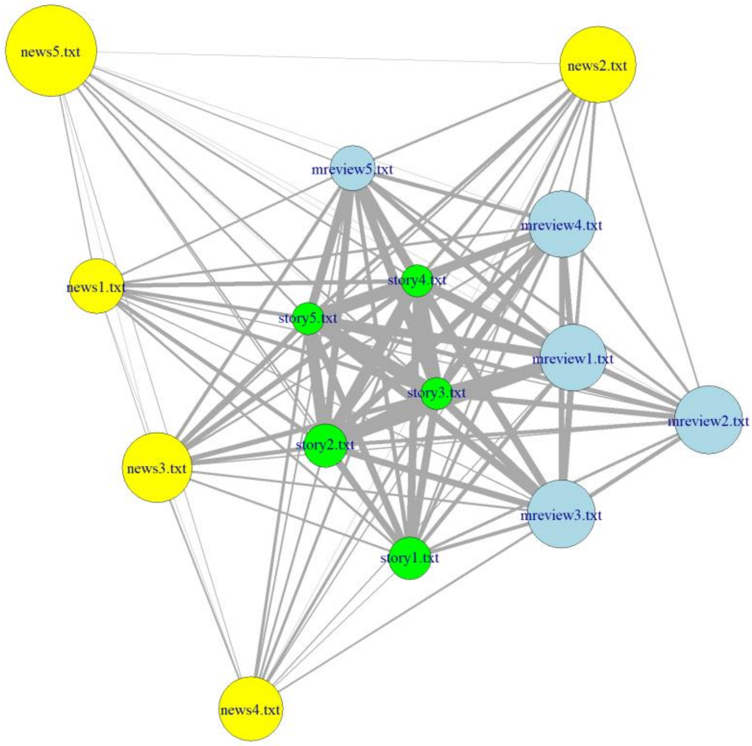
From the clustering accuracy shown above, we can see that both clustering method group the documents by their type pretty accurately, and the Cosine Distance Cluster performs slightly better than the Euclidean Distance Cluster.

Task 5

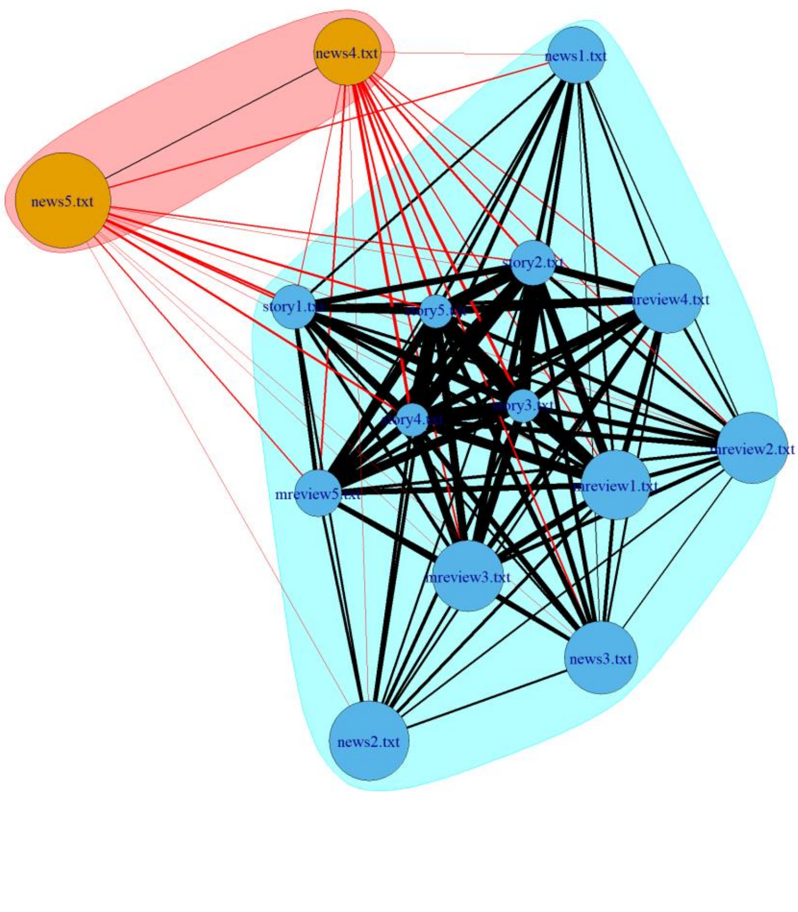
I have created a single-mode network showing the connections between the documents based on the number of shared terms. I have improved the basic network graph by showing some interesting features in it, the width of the edges represents the number of shared terms, and size of the vertex represents the closeness of the vertex. I have also represented different type of documents by different color and computed the community group in the graph using fast greedy algorithm. The summary of the eigenvector, betweenness, closeness and degree is computed and shown in the table below. (Refer to code at appendix[5])

	degree	closeness	betweeness	evector
mreview1.txt	14	0.02040816	0.0000000	0.7536870
mreview2.txt	14	0.02083333	0.0000000	0.4367273
mreview3.txt	14	0.02083333	0.0000000	0.6446419
mreview4.txt	14	0.02040816	0.0000000	0.6330213
mreview5.txt	14	0.01388889	0.0000000	0.6832693
news1.txt	13	0.01666667	0.5333333	0.3341981
news2.txt	13	0.02325581	7.7333333	0.3059236
news3.txt	14	0.02127660	1.4500000	0.4094931
news4.txt	14	0.01960784	15.7833333	0.2306285
news5.txt	14	0.02777778	72.9833333	0.1538723
story1.txt	14	0.01298701	0.0000000	0.5825835
story2.txt	14	0.01333333	0.0000000	0.8889965
story3.txt	14	0.00990099	0.0000000	1.0000000
story4.txt	14	0.00990099	0.0000000	0.9782974
story5.txt	14	0.00990099	0.0000000	0.9558059

From the table, we can see that most of the vertex has connection to every other vertex. All of the news documents has betweenness greater than 0, others all have betweenness 0. Story documents generally has higher value of eigenvector but lower value of closeness.


news5.txt
news2.txt
mreview5.txt
mreview4.txt
news1.txt
story4.txt
story5.txt
mreview1.txt
story3.txt
mreview2.txt
news3.txt
story2.txt
mreview3.txt
story1.txt
news4.txt

Fast Greedy


news4.txt
news1.txt
news5.txt
story2.txt
story1.txt
story5.txt
mreview4.txt
story4.txt
story3.txt
mreview2.txt
mreview5.txt
mreview1.txt
mreview3.txt
news3.txt
news2.txt

From the network graph, we can see that all the news document are on the edge of the graphs whilst all the story documents are in the centre of the graph. There is no clear group in the graph as almost all of the vertices are connected to each other. Another interesting finding is that the connections between story documents and the other documents are generally stronger.

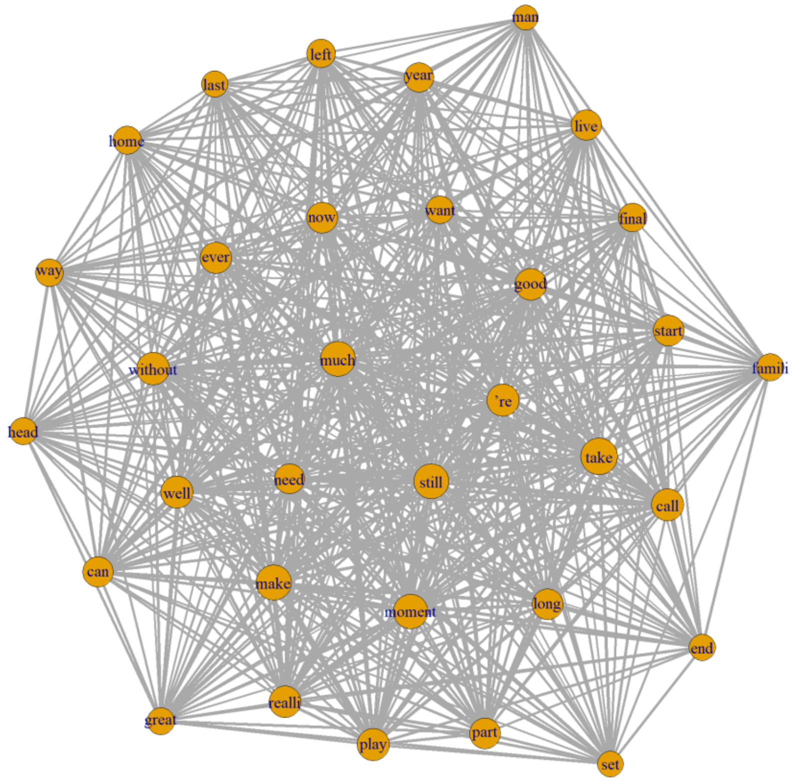
With the information above, we can deduce that the most important(central) documents in the network are the story documents as they have higher eigenvector and stronger connections to other vertices.

Task 6

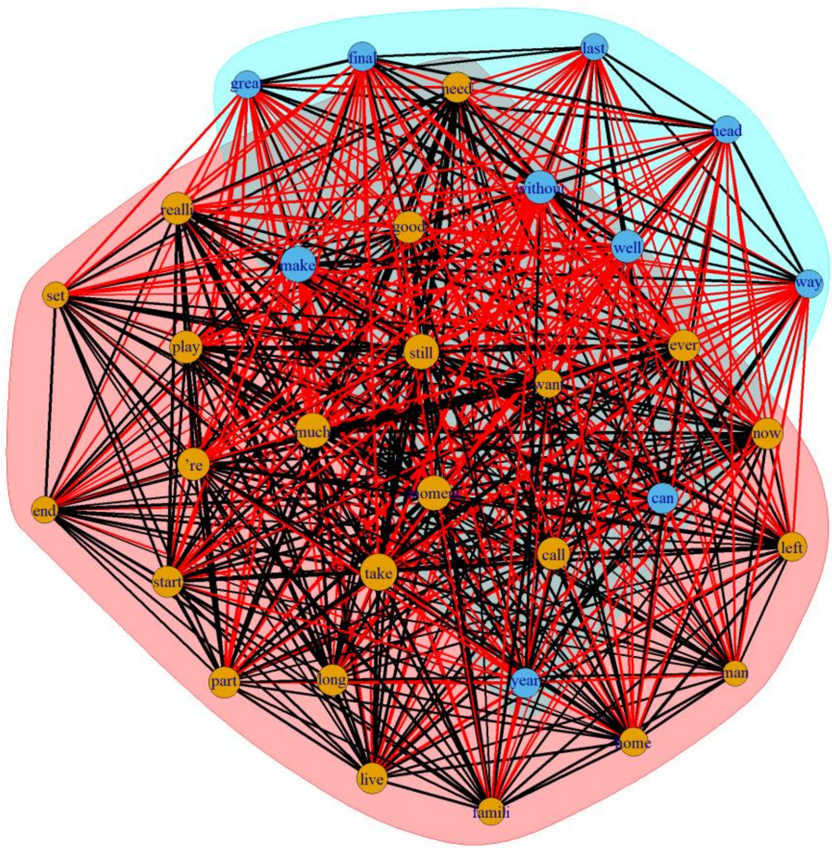
I have created a single-mode network showing the connections between the tokens. I have improved the basic network graph by showing some interesting features in it, the width of the edges represents the number of shared terms, and size of the vertex represents the eigenvector of the vertex. I have also computed the community group in the graph using fast greedy algorithm. The summary of the eigenvector, betweenness, closeness and degree is computed and shown in the table below. (Refer to code at appendix[6])

	degree	closeness	betweeness	evector
take	32	0.004385965	0.0000000	1.0000000
make	32	0.004608295	0.0000000	0.9653841
much	32	0.004651163	0.0000000	0.9496289
still	32	0.004672897	0.0000000	0.9437096
moment	32	0.004807692	0.0000000	0.9292496
without	32	0.004950495	0.0000000	0.8916913
call	32	0.005025126	0.0000000	0.8800916
play	32	0.005050505	0.0000000	0.8795403
’re	32	0.005050505	0.0000000	0.8795403
well	32	0.005050505	0.0000000	0.8779112
realli	32	0.005181347	0.0000000	0.8681957
good	32	0.005154639	0.0000000	0.8646580
ever	32	0.005181347	0.0000000	0.8549979
now	32	0.005263158	0.0000000	0.8404848
part	32	0.005291005	0.0000000	0.8378636
start	32	0.005291005	0.0000000	0.8378636
long	32	0.005319149	0.0000000	0.8350252
can	32	0.005319149	0.0000000	0.8308336
live	32	0.005405405	0.0000000	0.8200018
need	32	0.005405405	0.0000000	0.8193341
year	32	0.005494505	0.0000000	0.8067949
final	32	0.005681818	0.0000000	0.7782941
left	32	0.005747126	0.3333333	0.7757716
home	32	0.005780347	0.2500000	0.7689318
way	32	0.005917160	0.2500000	0.7498480
want	32	0.005882353	0.0000000	0.7497101
great	32	0.005952381	1.0000000	0.7463318
famili	32	0.005952381	1.1666667	0.7444659
head	32	0.005988024	1.0000000	0.7423339
end	32	0.006250000	3.1666667	0.7155164
last	32	0.006211180	0.2500000	0.7130260
set	32	0.006329114	5.8333333	0.7049063
man	32	0.006535948	11.6666667	0.6762068

From the table we can see that most of the tokens have betweenness of zero. All of them have degree of 32 which means all of them are connected to each other. The eigenvector value ranges between 0.67 and 1, and closeness ranges between 0.0043 and 0.0065. The higher the eigenvector, the lower the closeness.


man
left
year
last
live
home
final
now
want
ever
way
good
start
famili
without
much
're
head
take
well
need
still
call
can
make
moment
long
end
realli
great
play
part
set

Fast Greedy

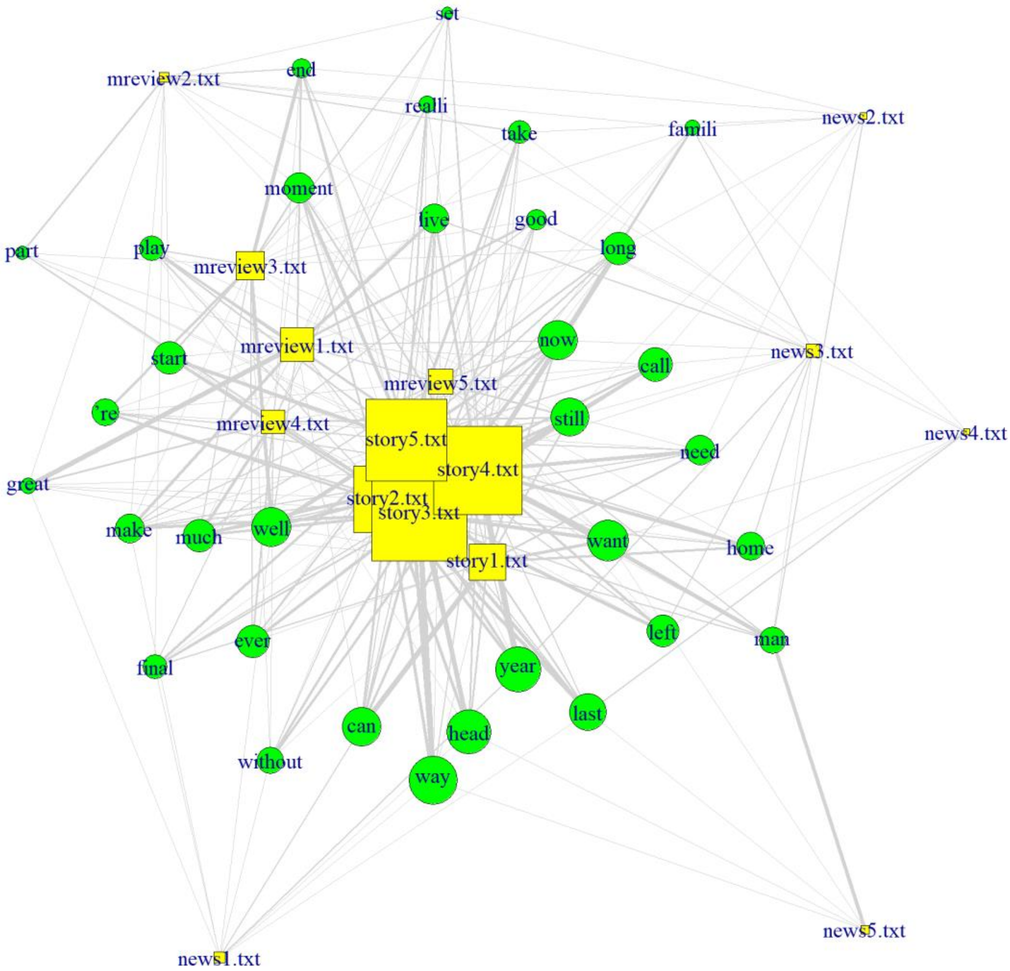

final
last
grea
need
head
ithou
realli
good
well
make
way
set
play
still
ever
wan
much
now
're
omen
can
end
left
call
take
start
part
long
year
man
ome
live
famili

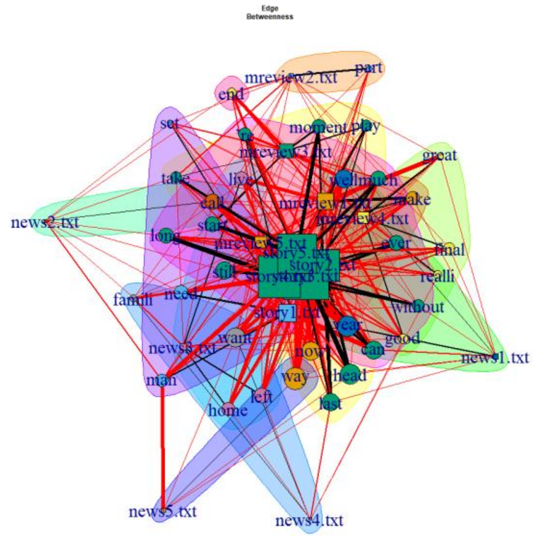
We can’t identify a clear group in the network graph, every node is connected to every other nodes and there’s no significant difference in between the strength of connections of the nodes. From the information above, we can say that the most important(central) tokens are ‘take’, ‘much’, ‘still’, ‘moment’ and ‘without’ because they have the greatest eigenvector value.

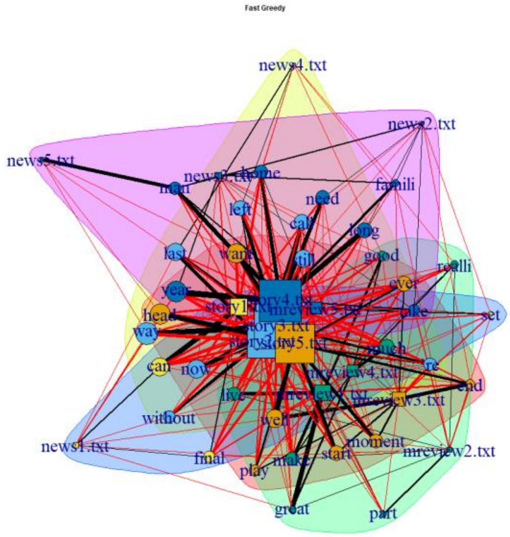
Task 7

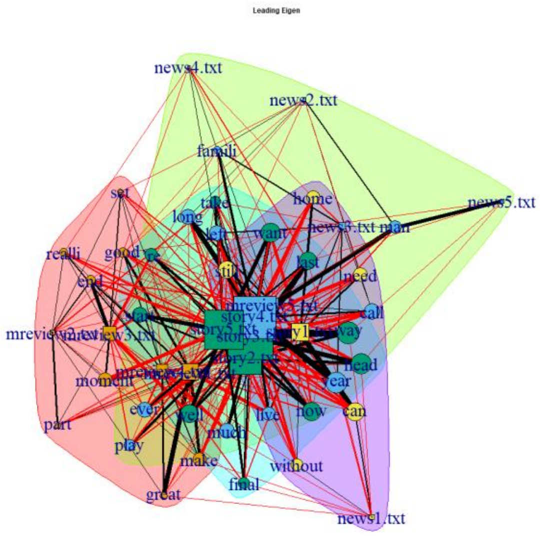
To create a bipartite network of my corpus, I first transformed my data into suitable format. And then I plotted the graph out with yellow square representing the documents, and green circle representing the tokens. To show the interesting features of the data more clearly, the width of each edge represents the strength of connection between the vertex. The size of the vertex represents the value of eigenvector of the vertex. I have also computed the community groups using 4 methods which are edge betweenness, fast greedy, leading eigen and label prop, all four of them are plotted out. The summary of the eigenvector, betweenness, closeness and degree is computed and shown in the table below. (Refer to code at appendix [7])

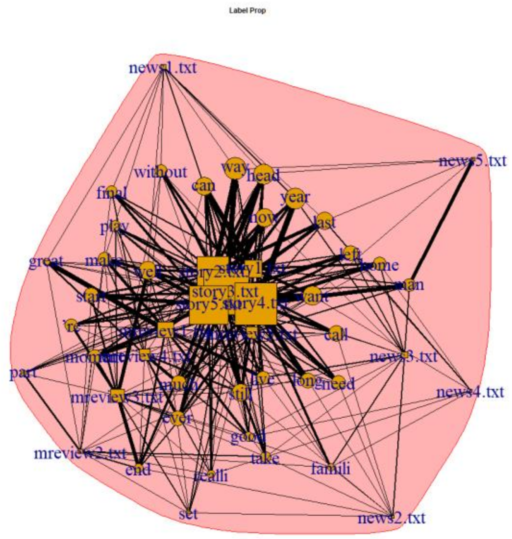
	degree	closeness	betweeness	evector
story3.txt	33	0.009090909	33.520910	1.00000000
story4.txt	32	0.009009009	19.611837	0.93579817
story5.txt	31	0.009523810	52.670042	0.86241422
story2.txt	28	0.009259259	41.947922	0.70665341
way	8	0.008695652	22.671835	0.51283994
year	8	0.008403361	1.620984	0.48052433
head	8	0.007633588	11.744501	0.47314833
want	9	0.009174312	31.720721	0.44143033
now	8	0.007246377	0.250000	0.42251587
well	9	0.008695652	15.333326	0.42031977
can	8	0.007936508	0.250000	0.41394942
still	10	0.008196721	5.108467	0.40466444
last	8	0.007462687	5.262488	0.39488941
story1.txt	18	0.009090909	44.000127	0.38587689
call	9	0.009708738	24.060184	0.36227578
mreview1.txt	23	0.010204082	135.823423	0.35342491
much	9	0.008771930	4.521611	0.35305191
ever	8	0.008403361	4.879417	0.35222455
long	8	0.008771930	8.352882	0.35135384
start	8	0.008695652	8.613284	0.35061864
left	8	0.008620690	7.252996	0.34474444
moment	9	0.009615385	17.563439	0.32547794
need	8	0.008695652	5.577009	0.31579971
make	10	0.009259259	14.035641	0.31262406
live	8	0.008333333	2.059608	0.31003466
home	8	0.008474576	5.090863	0.30607139
mreview3.txt	19	0.009708738	66.200680	0.29995308
’re	8	0.009174312	6.129510	0.29570577
man	8	0.008000000	2.378409	0.28928245
without	9	0.010204082	36.698417	0.28444017
play	8	0.008771930	5.091979	0.26510856
final	8	0.008620690	7.506196	0.26390527
mreview5.txt	21	0.010416667	132.881403	0.26321576
take	11	0.010101010	33.726708	0.25109961
mreview4.txt	19	0.009803922	96.441333	0.24935087
good	9	0.009090909	19.630773	0.22223795
end	8	0.008196721	4.452550	0.20972683
realli	9	0.009803922	26.278733	0.18197746
great	8	0.009259259	29.475269	0.17117979
famili	9	0.010000000	38.556475	0.16768316
part	8	0.009523810	22.491826	0.14673036
news3.txt	12	0.008849558	51.128210	0.14030895
set	8	0.009615385	20.508171	0.12750809
news1.txt	10	0.008849558	65.888242	0.11600892
mreview2.txt	13	0.009345794	63.824380	0.10074925
news5.txt	5	0.007407407	10.851979	0.08929766
news2.txt	9	0.008695652	34.408497	0.06948739
news4.txt	7	0.008547009	22.695287	0.06158782


set
mreview2.txt
end
realli
take
famili
news2.txt
moment
five
good
part
play
mreview3.txt
long
mreview1.txt
now
start
call
news3.txt
mreview5.txt
're
still
mreview4.txt
news4.txt
story5.txt
need
story4.txt
great
story2.txt
story3.txt
make
much
well
want
home
story1.txt
left
man
ever
final
year
last
can
head
without
way
news5.txt
news1.txt









From the plots and summary above, we can see that news type document has the least connection to the words, and story type documents have the greatest number of connections to the words and greater eigen vector value, hence we can say that they are the most important documents in the network. The review type documents have greatest betweenness. We are not able to identify any clear groups as all the community groups are all overlapped. The word ‘take’ has the most connections to the documents as it has the greatest number of degrees. The word ‘way’ is the most important(central) word as it has the greatest eigenvector value.

Task 8

To summarize all of my findings, through all the text network above, I have found that the most important documents among all 15 of them are story type documents as they generally have greater value of eigenvector. The most important tokens in my findings are ‘take’, ‘much’, ‘still’, ‘moment’ and ‘without’. For my dataset, clustering is effective in identifying different groups of documents as it has an accuracy more than 0.8. For the social network analysis, it performed relatively poorly as it is not able to show a clear group between the documents and tokens. However, it is able to show that which documents and tokens are relatively more important among the others.

Appendix

[1]

```
rm(list=ls())
library(slam)
library(tm)
library(SnowballC)

cname = file.path(".","txt")
print(dir(cname))
Corpus(DirSource((cname)))
```

[2]

```
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, stripWhitespace)
docs <- tm_map(docs, stemDocument, language = "english")
dtm <- DocumentTermMatrix(docs)
freq <- colSums(as.matrix(dtm))
length(freq)
ord <- order(freq)
freq[tail(ord,30)]



toSpace <- content_transformer(function(x, pattern) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "back")
docs <- tm_map(docs, toSpace, "work")
docs <- tm_map(docs, toSpace, "time")
docs <- tm_map(docs, toSpace, "mile")
docs <- tm_map(docs, toSpace, "like")
docs <- tm_map(docs, toSpace, "said")
docs <- tm_map(docs, toSpace, "just")
docs <- tm_map(docs, toSpace, "one")
docs <- tm_map(docs, toSpace, "’s")
```

```r
docs <- tm_map(docs, toSpace, "“")
docs <- tm_map(docs, toSpace, "will")
docs <- tm_map(docs, toSpace, "day")
docs <- tm_map(docs, toSpace, "get")
docs <- tm_map(docs, toSpace, "’m")
docs <- tm_map(docs, toSpace, "”")
docs <- tm_map(docs, toSpace, "know")
docs <- tm_map(docs, toSpace, "look")
docs <- tm_map(docs, toSpace, "made")
docs <- tm_map(docs, toSpace, "see")
docs <- tm_map(docs, toSpace, "gus")
docs <- tm_map(docs, toSpace, "never")
docs <- tm_map(docs, toSpace, "ask")
docs <- tm_map(docs, toSpace, "thing")
docs <- tm_map(docs, toSpace, "walk")
docs <- tm_map(docs, toSpace, "hand")
docs <- tm_map(docs, toSpace, "feel")
docs <- tm_map(docs, toSpace, "jevenust")
docs <- tm_map(docs, toSpace, "two")
docs <- tm_map(docs, toSpace, "’d")
docs <- tm_map(docs, toSpace, "thornton")

dtm <- DocumentTermMatrix(docs)
dtms <-removeSparseTerms(dtm, 0.47)
inspect(dtms)
dtms = as.matrix(dtms)
dtms
write.csv(dtms, "dtms.csv")
```

```
[1] 3026
    made    never      ask    thing     walk     hand     feel     even      two       ’d thornton      gus      see
      27       27       27       28       28       28       30       30       30       30       32       32       35
    will      day      get       ’m        ”     know     look     back     work     time     mile     like     said
      36       36       37       37       37       38       42       43       47       48       49       55       56
    just      one       ’s        “
      57       61       62       77
```

```
<<DocumentTermMatrix (documents: 15, terms: 33)>>
Non-/sparse entries: 280/215
Sparsity           : 43%
Maximal term length: 7
Weighting          : term frequency (tf)
Sample             :
              Terms
Docs           can head last much now still want way well year
  mreview1.txt   1    0    2    1   2     2    0   0    2    1
  mreview2.txt   0    0    0    0   0     0    1   0    0    0
  mreview3.txt   0    0    0    2   0     1    1   0    6    0
  mreview4.txt   0    1    0    5   0     3    0   0    1    0
  mreview5.txt   2    0    0    2   2     0    1   1    0    2
  story1.txt     8    3    0    1   3     3    5   1    0    0
  story2.txt     4    5    6    2   3     5    0   5    4    5
  story3.txt     4    4    6    4   6     4    6   9    3    4
  story4.txt     4    3    2    5   2     4    3   2    2   10
  story5.txt     2    8    3    1   5     2    7   6    7    1
```

```
              Terms
Docs           call can end good great last live long make moment much need now part play realli set start still
  mreview1.txt    1   1   3    4     8    2    6    1    5      1    1    1   2    1    2      1   1     1     2
  mreview2.txt    0   0   2    0     1    0    1    0    1      1    0    0   0    3    0      1   1     1     0
  mreview3.txt    0   0   6    1     0    0    0    1    1      3    2    0   0    1    2      1   2     3     1
  mreview4.txt    1   0   1    2     3    0    0    0    2      1    5    1   0    0    1      2   1     0     3
  mreview5.txt    1   2   2    0     0    0    1    3    2      1    2    0   2    0    1      0   0     0     0
  news1.txt       0   2   0    0     1    1    0    0    1      0    0    0   0    0    0      0   0     0     0
  news2.txt       1   0   1    0     0    0    0    1    0      0    0    1   0    0    0      0   1     0     1
  news3.txt       1   0   0    1     0    0    2    0    0      0    0    0   1    0    0      0   0     0     1
  news4.txt       0   0   0    1     0    2    0    0    0      0    0    0   0    0    0      1   0     0     0
  news5.txt       0   0   0    0     0    1    0    0    0      0    0    0   0    0    0      0   0     0     0
  story1.txt      0   8   0    0     1    0    0    0    0      0    1    3   3    1    0      0   2     1     3
  story2.txt      4   4   0    1     1    6    3    2    2      3    2    2   3    1    2      2   0     3     5
  story3.txt      6   4   1    3     1    6    2    1    5      1    4    2   6    2    1      1   2     4     4
  story4.txt      3   4   0    2     1    2    4    8    2      3    5    6   2    1    4      1   1     1     4
  story5.txt      2   2   4    1     0    3    2    3    1      6    1    2   5    1    3      3   0     6     2
              Terms
Docs           take well year 're famili final want ever without head left home man way
  mreview1.txt    1    2    1   1      0     0    0    0       0    0    0    0   0   0
  mreview2.txt    2    0    0   0      1     1    1    0       0    0    0    0   0   0
  mreview3.txt    1    6    0   4      1     0    1    2       1    0    0    0   0   0
  mreview4.txt    0    1    0   1      0     2    0    3       1    1    1    0   0   0
  mreview5.txt    2    0    2   1      1     0    1    1       1    0    1    1   2   1
  news1.txt       0    1    2   0      0     1    0    0       1    1    0    0   0   1
  news2.txt       1    0    0   0      1     0    0    0       0    0    0    0   2   0
  news3.txt       1    1    2   0      2     0    0    0       0    0    2    2   1   0
  news4.txt       0    0    0   0      1     0    1    0       0    0    1    1   0   0
  news5.txt       0    0    0   0      0     0    1    0       0    1    0    0   6   1
  story1.txt      1    0    0   0      0     2    5    1       1    3    0    4   2   1
  story2.txt      1    4    5   3      0     1    0    1       2    5    2    1   0   5
  story3.txt      2    3    4   4      1     3    6    5       4    4    6    2   2   9
  story4.txt      2    2   10   2      4     2    3    4       4    3    4    6   7   2
  story5.txt      4    7    1   2      1     4    7    3       1    8    2    2   1   6
```

[3] library(proxy)

k=3

distance_matrix <- dist(scale(dtms), method = "cosine")

fit = hclust(distance_matrix, method = "ward.D")

plot(fit, main = "Cosine Distance Cluster",hang = -1,xlab="Document")

cutihfit1 = cutree(fit, k = k)

rect.hclust(fit, k = k, border = "red")

```
area =
c("Reviews","Reviews","Reviews","Reviews","Reviews","News","News","News","News","
News","Stories","Stories","Stories","Stories","Stories")

table(GroupNames = area, Clusters = cutihfit1)
```

[4]

```
k=3

distance_matrix <- dist(scale(dtms))

fit = hclust(distance_matrix, method = "ward.D")

plot(fit, main = "Euclidean Distance Cluster",hang = -1,xlab="Document")

cutihfit2 = cutree(fit, k = k)

rect.hclust(fit, k = k, border = "red")

table(GroupNames = area, Clusters = cutihfit2)
```

[5]

```
png(filename="d1.png", width=2000, height=2000)

dtmsx <- as.matrix((dtms>0)+0)

byAbsMatrix <-dtmsx%*%t(dtmsx)

diag(byAbsMatrix)=0

byAbs = graph_from_adjacency_matrix(byAbsMatrix,mode = "undirected",weighted
=TRUE)

E(byAbs)$width = E(byAbs)$weight

V(byAbs)$size = closeness(byAbs)*1000

V(byAbs)$label.cex = 3

V(byAbs)$color=c("lightblue","lightblue","lightblue","lightblue","lightblue","yellow","yello
w","yellow","yellow","yellow","green","green","green","green","green")

plot(byAbs)

dev.off()


png(filename="d4.png", width=2000, height=2000)

ceb = cluster_fast_greedy(as.undirected(byAbs))

g_ceb = plot(ceb,

as.undirected(byAbs),vertex.label=V(byAbs)$role,main="Fast Greedy",cex.main=100)

dev.off()

a=as.table(degree(byAbs))

b=as.table(closeness(byAbs), digits = 2)


c=as.table(betweenness(byAbs))
```

```
d = as.table(evcent(byAbs)$vector)
stats = as.data.frame(rbind(a,b,c,d))
stats = as.data.frame(t(stats))
colnames(stats) = c("degree", "betweenness", "closeness", "eigenvector")
print.data.frame(stats)
```

[6]

```
png(filename="d2.png", width=2000, height=2000)
dtmsx = as.matrix((dtmsx>0)+0)
byTokenMatrix <-t(dtmsx)%*%dtmsx
diag(byTokenMatrix)=0
byToken = graph_from_adjacency_matrix(byTokenMatrix,mode = "undirected",weighted
=TRUE)
E(byToken)$width = E(byToken)$weight
V(byToken)$size = evcent(byToken)$vector*10
V(byToken)$label.cex = 3
plot(byToken)
dev.off()

png(filename="d5.png", width=2000, height=2000)
ceb = cluster_fast_greedy(as.undirected(byToken))
g_ceb = plot(ceb,
as.undirected(byToken),vertex.label=V(byToken)$role,main="Fast Greedy")
dev.off()

a=as.table(degree(byToken))
b=as.table(closeness(byToken), digits = 2)
c=as.table(betweenness(byToken))
d = as.table(evcent(byToken)$vector)
stats = as.data.frame(rbind(a,b,c,d))
stats = as.data.frame(t(stats))
colnames(stats) = c("degree",'closeness','betweeness','evector')
ord = order(-stats$evector)
print.data.frame(stats[ord,])
```

[7]

```
png(filename="d3.png", width=2000, height=2000)
```

```
g <- graph.data.frame(dtmsc, directed=FALSE)
bipartite.mapping(g)
V(g)$type <- bipartite_mapping(g)$type
V(g)$color <- ifelse(V(g)$type, "green", "yellow")
V(g)$shape <- ifelse(V(g)$type, "circle", "square")
E(g)$color <- "lightgray"
E(g)$width <-E(g)$weight
V(g)$size <- evcent(g)$vector*20
V(g)$label.cex = 3

plot(g)
dev.off()

png(filename="d6.png", width=2000, height=2000)
par(mfrow = c(2,2))
ceb = cluster_edge_betweenness(as.undirected(g))
g_ceb = plot(ceb,
as.undirected(g),vertex.label=V(g)$role,main="Edge
Betweenness")
ceb = cluster_fast_greedy(as.undirected(g))
g_ceb = plot(ceb,
as.undirected(g),vertex.label=V(g)$role,main="Fast Greedy")
ceb = cluster_leading_eigen(as.undirected(g))
g_ceb = plot(ceb,
as.undirected(g),vertex.label=V(g)$role,main="Leading Eigen")
ceb = cluster_label_prop(as.undirected(g))
g_ceb = plot(ceb,
as.undirected(g),vertex.label=V(g)$role,main="Label Prop")
dev.off()

a=as.table(degree(g))
b=as.table(closeness(g), digits = 2)
```

```
c=as.table(betweenness(g))

d = as.table(evcent(g)$vector)

stats = as.data.frame(rbind(a,b,c,d))

stats = as.data.frame(t(stats))

colnames(stats) = c("degree",'closeness','betweeness','evector')

ord = order(-stats$evector)

print.data.frame(stats[ord,])
```

References

Anderson, A. (2023a) *Coward's punch victim dies in hospital - news.com.au, Gold Coast coward's punch victim dies in hospital after weekend assault*. Available at: https://www.news.com.au/national/queensland/crime/gold-coast-cowards-punch-victim-dies-in-hospital-after-weekend-assault/news-story/5b7c82da67dab342050acc70969afe66 (Accessed: 09 June 2023).

Anderson, A. (2023b) *Man charged over mother-of-three's third-degree car fire burns*. Available at: https://www.news.com.au/technology/motoring/on-the-road/motherofthree-suffered-thirddegree-burns-in-car-fire-before-explosion/news-story/a1adc755f378f49f4ce8231ff15e01a8 (Accessed: 09 June 2023).

Beatty, L. (2023) *Schoolboy charged after fatal stabbing of Pa Sawm Lyhym in Melbourne ..., Schoolboy charged after fatal stabbing of Pa Sawm Lyhym in Melbourne*. Available at: https://www.news.com.au/national/victoria/crime/schoolboy-charged-after-fatal-stabbing-of-pa-sawm-lyhym-in-melbourne/news-story/a1774550a31480ababbece68bf8c7941 (Accessed: 09 June 2023).

Brwc (2023a) *Fast X - the BRWC review - film reviews, interviews, features: BRWC, film reviews, interviews, features | BRWC*. Available at: https://battleroyalewithcheese.com/2023/05/fast-x-the-brwc-review/ (Accessed: 09 June 2023).

Brwc (2023b) *The little mermaid: The BRWC review - film reviews, interviews, features: BRWC, film reviews, interviews, features | BRWC*. Available at: https://battleroyalewithcheese.com/2023/05/the-little-mermaid-the-brwc-review/ (Accessed: 09 June 2023).

Cook, L. (2023) *Cook review: 'across the spider-verse' crawls with action, mind-boggling animation, WHBF - OurQuadCities.com*. Available at: https://www.ourquadcities.com/entertainment-news/movies/cook-review-across-the-spider-verse-crawls-with-action-mind-boggling-animation/ (Accessed: 09 June 2023).

Delfosse, A. (2023) *Hazel – a fiction short story by Anne Delfosse – reedsy prompts, Reedsy*. Available at: https://blog.reedsy.com/short-story/24629v/ (Accessed: 09 June 2023).

George, S. (2023) *To plant a garden. – a science fiction short story by Scott George – Reedsy prompts, Reedsy*. Available at: https://blog.reedsy.com/short-story/gnppgd/ (Accessed: 09 June 2023).

Henkel, E. (2022) *All I want for christmas is fair labor practices – A christmas short story by Elizabeth Henkel – reedsy prompts, Reedsy*. Available at: https://blog.reedsy.com/short-story/9wv9wh/ (Accessed: 09 June 2023).

Laura (2023) *Past lives, Reeling Reviews*. Available at: https://www.reelingreviews.com/reviews/past-lives/ (Accessed: 09 June 2023).

McMillan, I. (2023) *Father, son found dead in their home in suspected murder-suicide*. Available at: https://www.news.com.au/national/nsw-act/father-and-son-found-dead-in-their-home-with-gunshot-wounds/news-story/78c70cf0d3bafaad24ac021732a23b54 (Accessed: 09 June 2023).

Oliver, M. (2023) *All the lonely people – a romance short story by Michelle Oliver – reedsy prompts, Reedsy*. Available at: https://blog.reedsy.com/short-story/kuqrmp/ (Accessed: 09 June 2023).

Sarah (2023) *Review: Guardians of the galaxy vol 3 is satisfying conclusion to Marvel's weirdest franchise*, *Review: Guardians of the Galaxy Vol 3 is satisfying conclusion to Marvel's weirdest franchise*. Available at: https://www.laineygossip.com/review-guardians-of-the-galaxy-vol-3-is-satisfying-conclusion-to-marvels-weirdest-franchise/74025 (Accessed: 09 June 2023).

Saunokonoko, M. (2023) *'so elated it's not funny': Kathleen Folbigg speaks following pardon*, *Kathleen Folbigg pardon: Update as mother freed after 20 years in jail*. Available at: https://www.9news.com.au/national/kathleen-folbigg-pardoned-after-20-years-in-jail/1973d88d-63c3-4f3e-bc8c-bf328357d2ff (Accessed: 09 June 2023).

Walker, A. (2023) *Out of place on the Appalachian Trail – a adventure short story by Aeris Walker – reedsy prompts*, *Reedsy*. Available at: https://blog.reedsy.com/short-story/0ik4ap/ (Accessed: 09 June 2023).

DTM table

	call	can	end	good	great	last	live	long	make
mreview1.txt	1	1	3	4	8	2	6	1	5
mreview2.txt	0	0	2	0	1	0	1	0	1
mreview3.txt	0	0	6	1	0	0	0	1	1
mreview4.txt	1	0	1	2	3	0	0	0	2
mreview5.txt	1	2	2	0	0	0	2	3	2
news1.txt	0	2	0	0	1	1	0	0	1
news2.txt	1	0	1	0	0	0	0	1	0
news3.txt	1	0	0	1	0	0	2	0	0
news4.txt	0	0	0	1	0	2	0	0	0
news5.txt	0	0	0	0	0	1	0	0	0
story1.txt	0	8	0	0	1	0	0	0	0
story2.txt	4	4	0	1	1	6	3	2	2
story3.txt	6	4	1	3	1	6	2	1	5
story4.txt	3	4	0	2	1	2	4	8	2
story5.txt	2	2	4	1	0	3	2	3	1

	moment	much	need	now	part	play	realli	set	start
mreview1.txt	1	1	1	2	1	2	1	1	1
mreview2.txt	1	0	0	0	3	0	1	1	1
mreview3.txt	3	2	0	0	1	2	1	2	3
mreview4.txt	1	5	1	0	0	1	2	1	0
mreview5.txt	1	2	0	2	0	1	0	0	0
news1.txt	0	0	0	0	0	0	0	0	0
news2.txt	0	0	1	0	0	0	0	1	0
news3.txt	0	0	0	1	0	0	0	0	0
news4.txt	0	0	0	0	0	0	1	0	0
news5.txt	0	0	0	0	0	0	0	0	0
story1.txt	0	1	3	3	1	0	0	2	1
story2.txt	3	2	2	3	1	2	2	0	3
story3.txt	1	4	2	6	2	1	1	2	5
story4.txt	3	5	6	2	1	4	1	1	1
story5.txt	6	1	2	5	1	3	3	0	6

	still	take	well	year	â€™re	famili	final	want	ever
mreview1.txt	2	1	2	1	1	0	0	0	0
mreview2.txt	0	2	0	0	0	1	1	1	0
mreview3.txt	1	1	6	0	4	1	0	1	2
mreview4.txt	3	0	1	0	1	0	2	0	3
mreview5.txt	0	2	0	2	1	1	0	1	1
news1.txt	0	0	1	2	0	0	1	0	0
news2.txt	1	1	0	0	0	1	0	0	0
news3.txt	1	1	1	2	0	2	0	0	0
news4.txt	0	0	0	0	0	1	0	1	0
news5.txt	0	0	0	0	0	0	0	1	0

story1.txt	3	1	0	0	0	0	2	5	1
story2.txt	5	1	4	5	3	0	1	0	1
story3.txt	4	2	3	4	4	1	3	6	5
story4.txt	4	2	2	10	2	4	2	3	4
story5.txt	2	4	7	1	2	1	4	7	3

	without	head	left	home	man	way
mreview1.txt	0	0	0	0	0	0
mreview2.txt	0	0	0	0	0	0
mreview3.txt	1	0	0	0	0	0
mreview4.txt	1	1	1	0	0	0
mreview5.txt	1	0	1	1	2	1
news1.txt	1	1	0	0	0	1
news2.txt	0	0	0	0	2	0
news3.txt	0	0	2	2	1	0
news4.txt	0	0	1	1	0	0
news5.txt	0	1	0	1	6	1
story1.txt	1	3	0	4	2	1
story2.txt	2	5	2	1	0	5
story3.txt	4	4	6	2	2	9
story4.txt	4	3	4	6	7	2
story5.txt	1	8	2	2	1	6