JuneJin_31994695.pdf

Name : June Jin | Student ID : 31994695

Question 1.

Collect a set of (machine-readable text) documents from an area of interest.

First, I made all the website to pdf from the CNN website (total 15 news / articles).

In World Topic

1. Singapore Airline Accident
2. South Koreans compete to see who doing absolutely nothing
3. In the world’s biggest election, millions of migrants are unable to vote
4. Mexican drug gangs ‘increasingly targeting’ Australia as meth supplies overtake rivals, police say
5. London’s famous Garrick Club votes to allow women, nearly 200 years after it was founded

In Food Topic

1. This tiny taco stand in Mexico has just earned a Michelin star
2. I tried gourmet food prepared from chicken feathers. Here’s how it’s made – and what it tasted like
3. Paris waiters compete in race to get a coffee and croissant across the capital
4. Inside Tokyo’s oldest onigiri restaurant
5. What is an IPA? A deliciously happy accident of beer history or the colonial marketing of a frugal recipe?

In Tech Topic

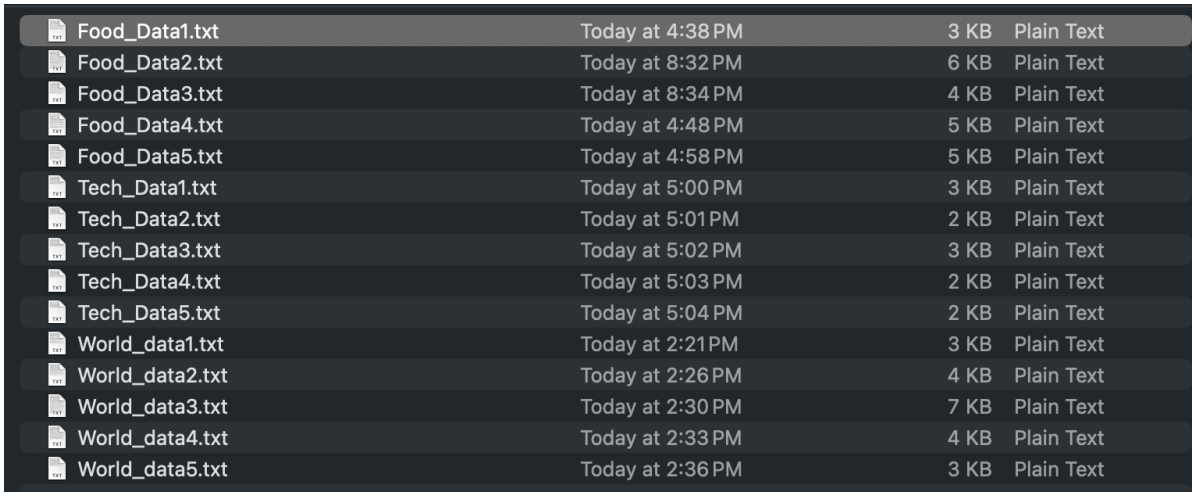
1. Facebook and Instagram probed over fears they may be too addictive for children

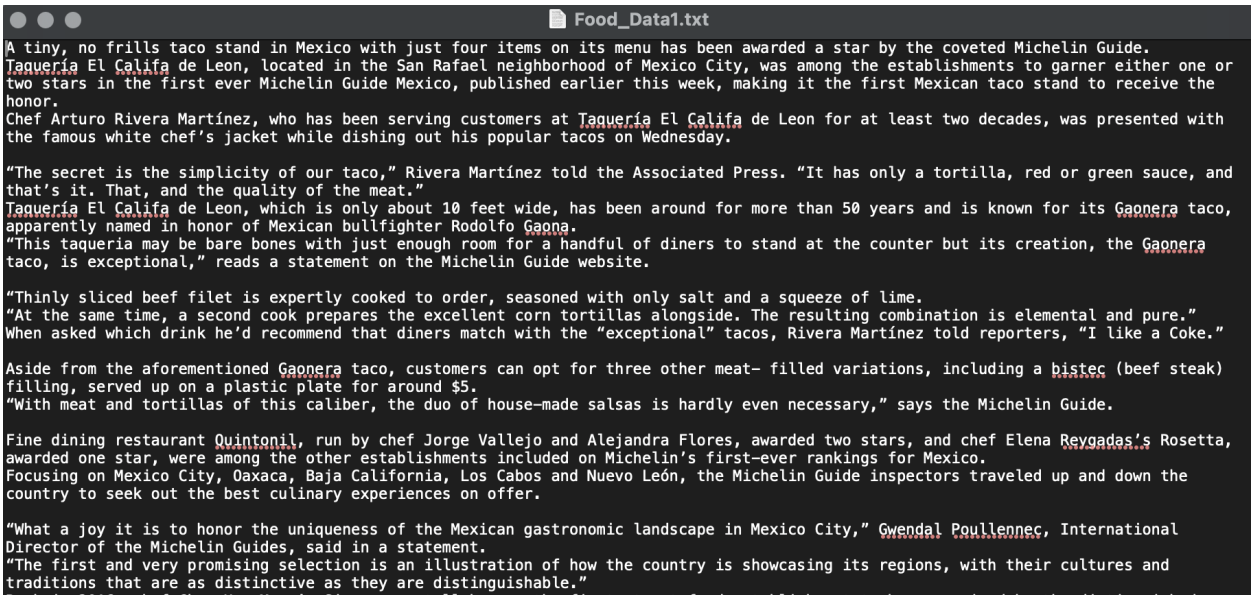
2. Elon Musk says AI will take all our jobs
3. Amazon Web Services CEO to step down
4. Microsoft asks some employees in China to move to other countries
5. FBI warns consumers not to use public phone charging stations

Question 2.

Create your corpus by first converting each document into a text format.

I converted all the documents to PDF format and used "TextEdit" on my Mac to transfer the text from the PDFs to TXT files. I then created a separate file called Data_text within the Assignment 3 folder to store the data.





Question 3.

Creating my Document-Term Matrix (DTM)

I removed the numbers which are considered noise in the text analysis with "removeNumbers".

I removed the punctuations because it is not related with text analysis by using "removePunctuation".

I converted all the text to lowercase for match formatting by using "content_transformer(tolower)".

I removed some words which is not important, and it makes more confuse for analyzing the data. For example, "'s", "'ve", "-" and so on. I made the function which is called "juneremoveChars", it will help us to find that word we want to remove.

Also, I removed all the Stopwords ("the", "and", "in", "of" and so on) by using stopwords("english")

At first before removing the sparse words, the dim was (15, 2090), which means 2090 terms. After applying removing sparse words by using "removeSparseTerms", it became to (15,12) which means 12 terms.

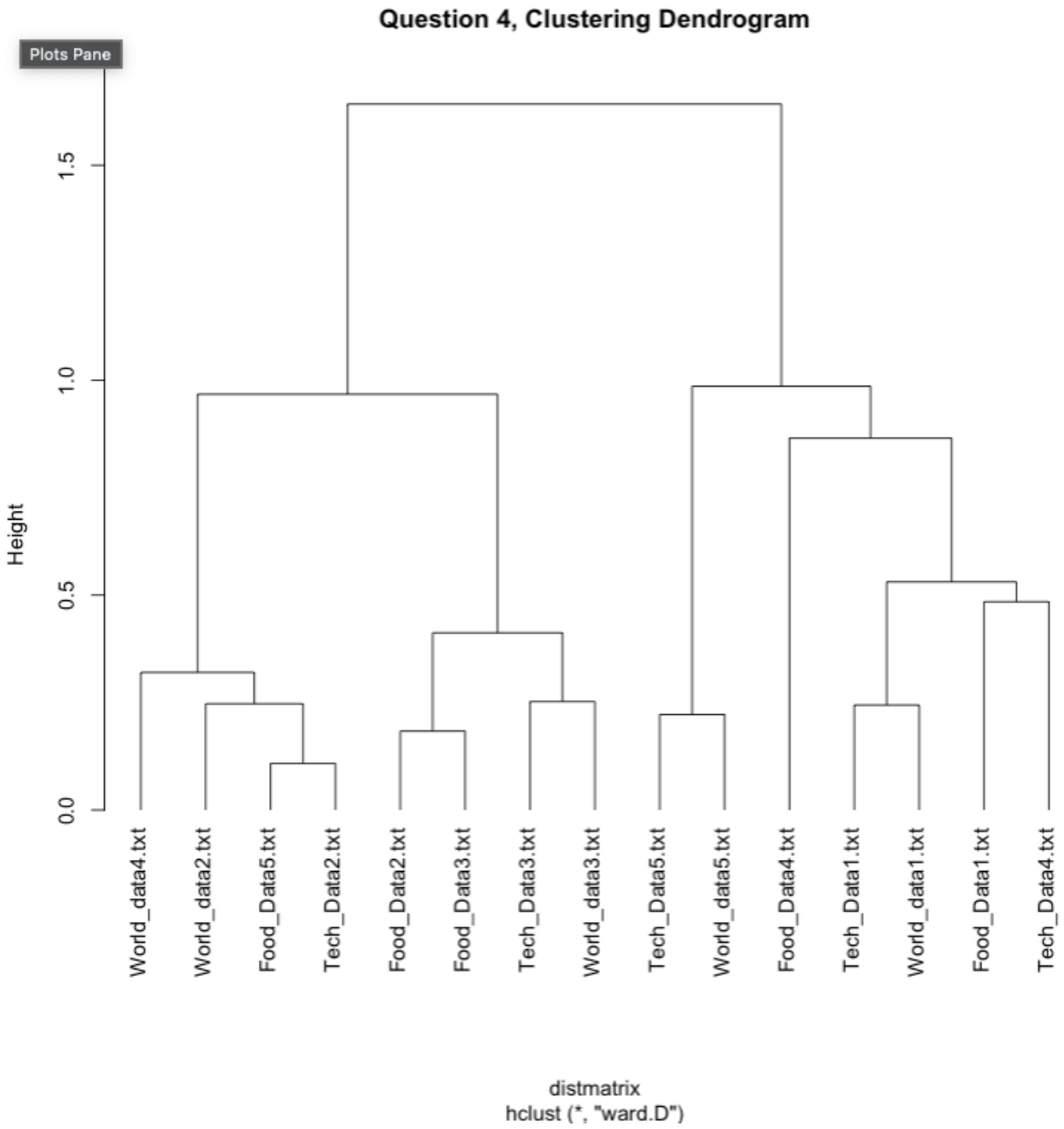
Question 4.

Create a hierarchical clustering of your corpus and show this as a dendrogram.

I used cosine distance to cluster each text file for create hierarchical clustering of my corpus.

Hierarchical clustering was executed to minimize the overall within-cluster variance. The dendrogram was partitioned into 15 clusters, matching the number of documents, with each document assigned a cluster label.

To quantitatively evaluate the clustering's efficacy, I initially compiled a list of topics or types of instruments relevant to the corpus. Subsequently, I plotted the cluster table and computed the accuracy.




```
      2 1 3
food  3 2 0
tech  2 2 1
world 3 1 1
> TA_matrix <- as.matrix(TA)
> diag_TA <- diag(TA_matrix)
> accuracy <- sum(diag_TA) / sum(TA_matrix)
> accuracy
[1] 0.4
```

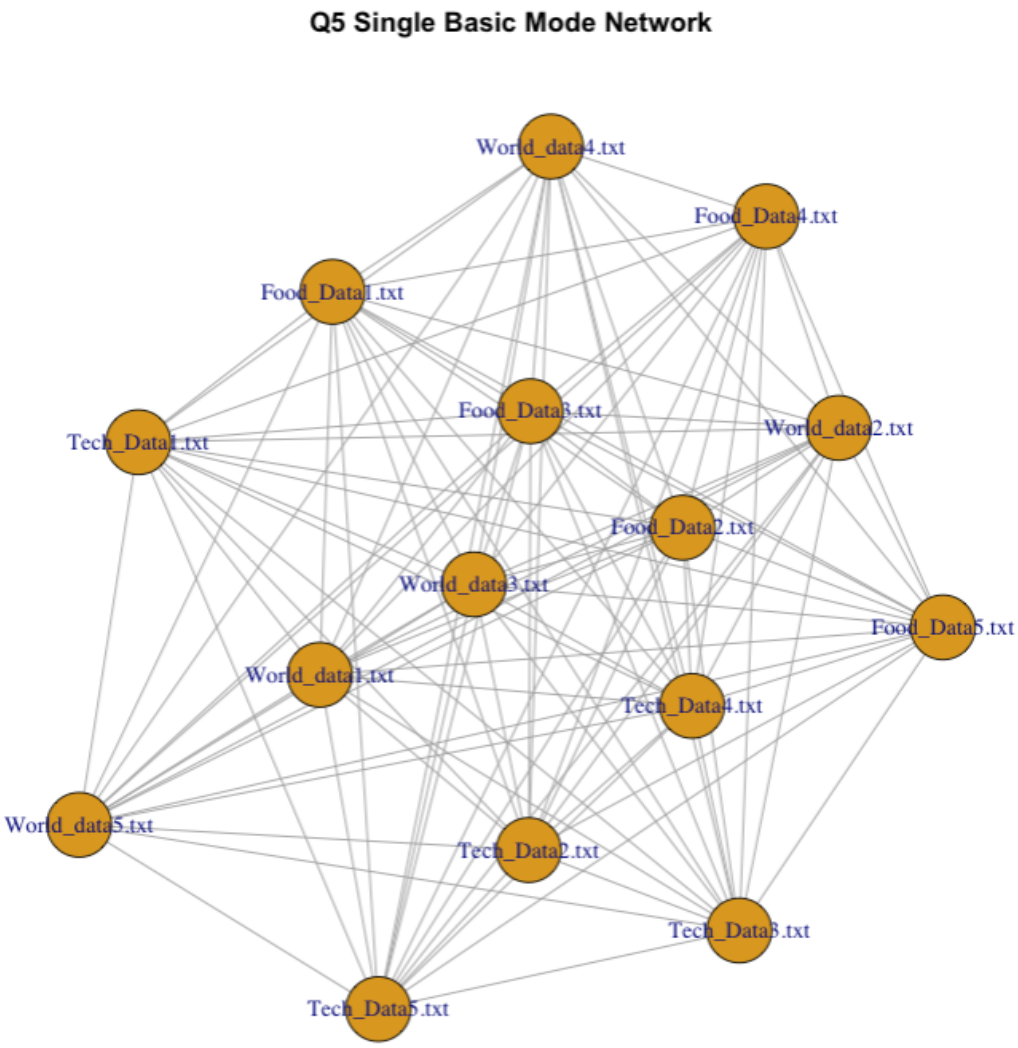
The accuracy of the clustering table was only 0.4, which means, only half and a bit of data matches with the actual cluster.

Question 5.

Creating a single-mode network showing the connection between documents.

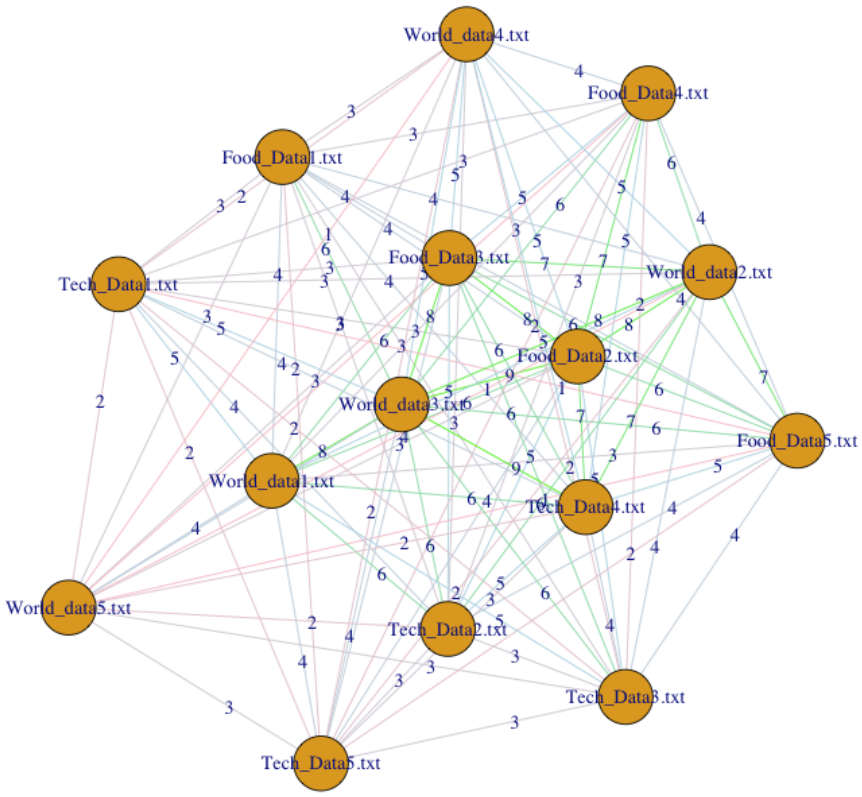
1. Calculation connections between each document using method.
    a. Convert the distance matrix to binary matrix
    b. multiply binary matrix by its transpose
    c. make leading diagonal zero

This is the basic single-mode network plot below :



The current plot does not effectively highlight the key relationships between the documents. Enhancing the visualization by coloring the edges according to their weights can improve clarity.

I tried to improve my adding the weight of between documents, and put the different colors based on how weight is strong or weak between them.



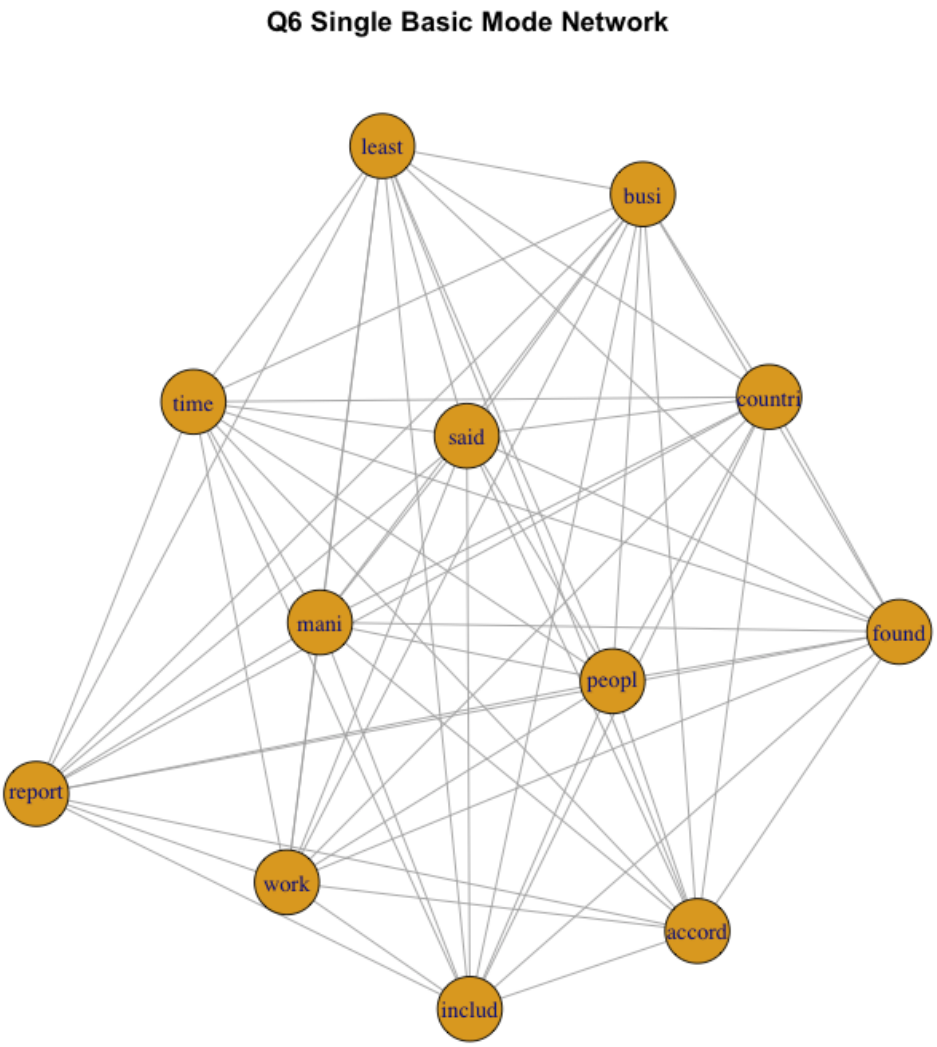
As shown in the plot above, the edges with the strongest weights are colored green, followed by light blue for moderate weights, and pink for the weakest weights. By using different colors for the edges, we can easily observe a distinct cluster in the center of the network. Additionally, this method helps us identify the outliers located at the periphery of the network, where the edges are colored red and light blue.

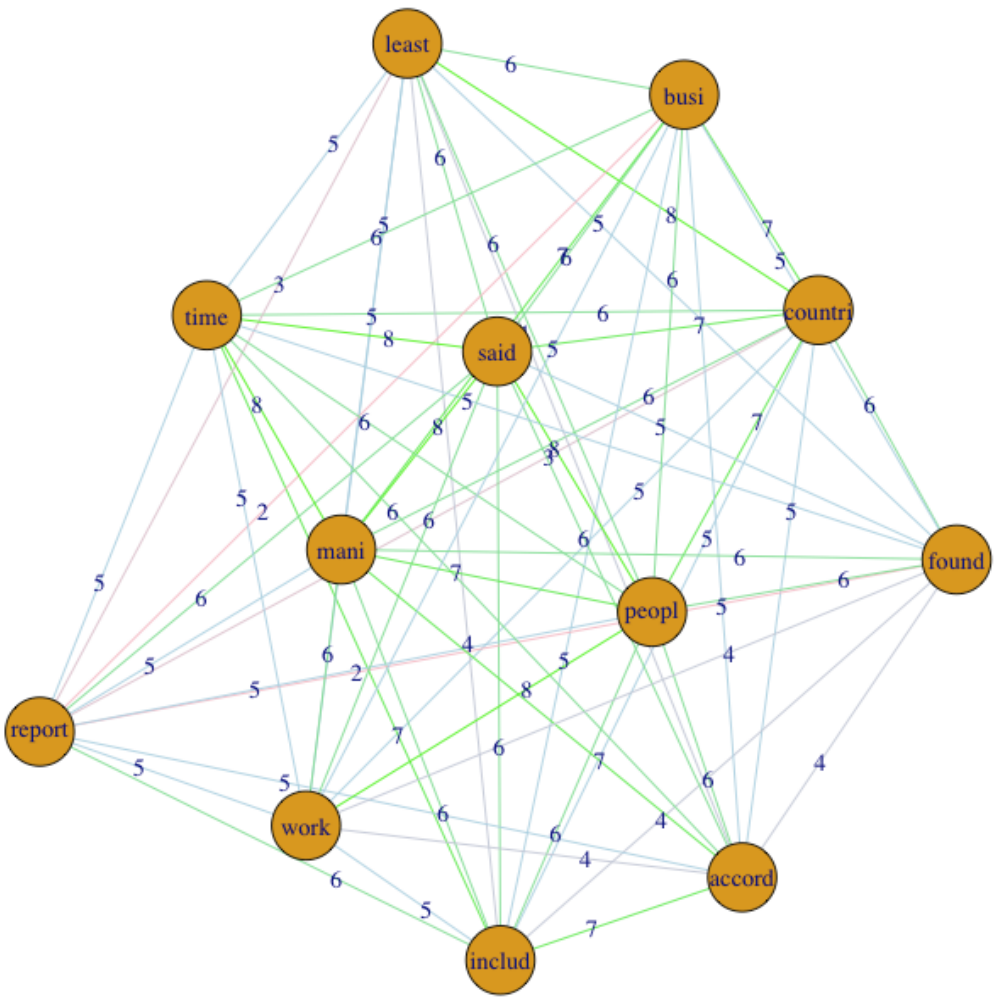
The node positioned at the topmost part of the network has edges with low weights and is situated farther from the network's core, indicating a weak connection with the other documents. Conversely, the majority of edges in the network's center are green, signifying that these documents are closely interconnected, sharing common terms frequently used in Food and World reviews.

Question 6.

Creating a single-mode network showing the connection between words.

the progression of this question 6 is same way with question 5.





Based on the final network plot, we can also identify the nodes which are important by looking at the color of the edges. Overall, by manually looking at the meaning of the nodes, the network doesn't quite well in predicting the important words used in the article. For example, commonly used words for reviewing the "world", "tech", and "food" were overlapped. But we could see some strong relationship such as "time" and "many", "said" and so on.

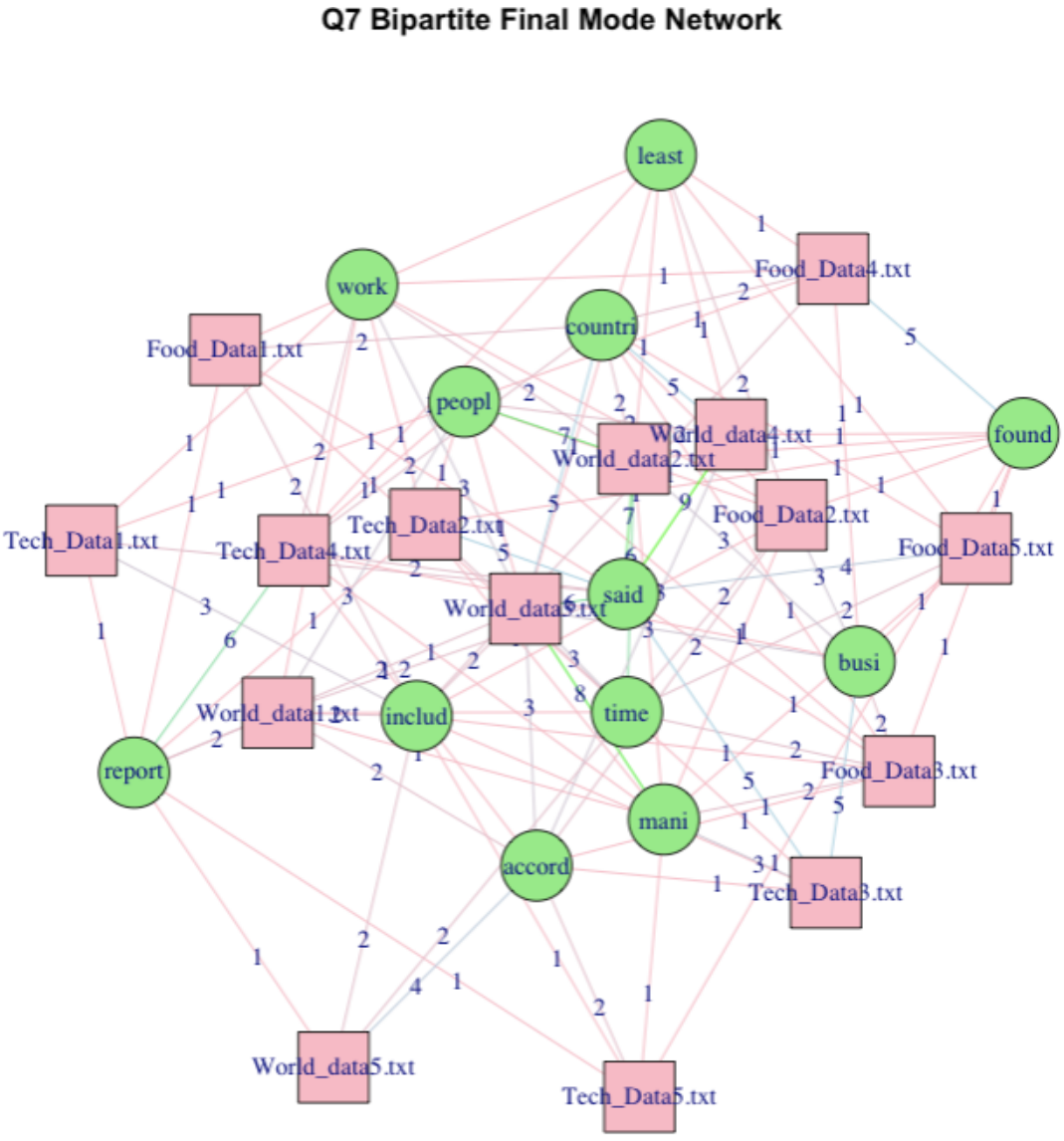
Question 7

a bipartite (two-mode) network of corpus, with document ID

Initially, a table is created listing the document names, tokens, and their respective weights. The table is shown below:

```
> dtmsc
                abs   token weight
1    Food_Data1.txt countri      2
2    Food_Data1.txt  includ      2
3    Food_Data1.txt   least      1
4    Food_Data1.txt  report      1
5    Food_Data1.txt    said      1
6    Food_Data1.txt    time      1
13   Food_Data2.txt countri      1
14   Food_Data2.txt  includ      1
15   Food_Data2.txt   least      2
18   Food_Data2.txt    time      2
19   Food_Data2.txt  accord      2
20   Food_Data2.txt    busi      3
21   Food_Data2.txt   found      1
22   Food_Data2.txt    mani      1
23   Food_Data2.txt   peopl      1
24   Food_Data2.txt    work      1
25   Food_Data3.txt countri      1
26   Food_Data3.txt  includ      1
29   Food_Data3.txt    said      1
30   Food_Data3.txt    time      2
31   Food_Data3.txt  accord      1
32   Food_Data3.txt    busi      2
33   Food_Data3.txt   found      1
34   Food_Data3.txt    mani      2
35   Food_Data3.txt   peopl      1
37   Food_Data4.txt countri      2
38   Food_Data4.txt  includ      2
```

Through this table, we can assign different colors (i will use the same color with Q5&6, and shapes to the types of nodes when we try to make the bipartite network.



As shown in the plot above, the pink square nodes represent the documents, while the light green circles represent the words. The edges are colored according to their weights, indicating the strength of the relationship between the documents and words.

From the graph, we can observe distinct clusters of documents. For example, in the center, documents like world2, world4, people, and country form a cluster due to their close proximity. Notably, world2 has a strong connection with the words “people” and “said,” with a weight of 7. Similarly, Tech4, Tech2, and Tech1 are also part of this cluster, with Tech4 having a particularly strong relationship with the

word “report.” Additionally, Food2, Food5, Food3, and Tech3 appear to form a cluster around the word “busy,” which is commonly used in restaurant-related articles.

In my opinion, the graph did not effectively form distinct clusters for Tech, World, and Food articles and their related words. This may be because these articles share many commonly used words regardless of their subject matter. For instance, articles about the world often include words like country and people, while tech and food articles also have their own common vocabulary. This results in the word “said” being centrally clustered with many articles.

In conclusion, text analysis can be a valuable tool for examining relationships between text-based documents, identifying key similarities and differences, and understanding how individual words relate to each other and to the documents.

Appendix

References for Question 1

Alberti, M., & CNN. (2024, May 23).
More than 20 people hurt on turbulent Singapore Airlines flight have spinal injuries, hospital says. CNN. https://www.cnn.com/2024/05/23/asia/singapore-airlines-flight-spinal-injuries-intl
Allan, D. (2024, April 7).
What is an IPA? A deliciously happy accident of beer history or the colonial marketing of a frugal recipe? CNN. https://www.cnn.com/travel/what-is-an-ipa
Cassanelli, J. Y., Noemi. (2024, May 14).
In the world’s biggest election, millions of migrants are unable to vote. CNN. https://www.cnn.com/2024/05/13/india/india-elections-migrant-workers-voting-intl-hnk-dst
Collins, L. (2024, April 12).
I tried gourmet food prepared from chicken feathers. Here’s how it’s made – and what it tasted like. CNN. https://www.cnn.com/2024/04/12/travel/kera-chicken-feather-protein-tasting-c2e-spc
Duffy, C. (2024, May 14).

Amazon Web Services CEO to step down | CNN Business. CNN. https://www.cnn.com/2024/05/14/tech/aws-ceo-adam-selipsky-steps-down
Guy, J. (2024, May 8). *London's famous Garrick Club votes to allow women, nearly 200 years after it was founded*. CNN. https://www.cnn.com/2024/05/08/uk/garrick-club-votes-women-gbr-intl-scli
Hardingham-Gill, T. (2024, May 16). *This tiny taco stand in Mexico City has just earned a Michelin star*. CNN. https://www.cnn.com/travel/taqueria-el-califa-de-leon-mexico-city
Jeong, S. (2024, May 17). *South Koreans compete to see who's best at doing absolutely nothing*. CNN. https://www.cnn.com/2024/05/16/asia/south-korea-space-out-competition-intl-hnk
Kelly, S. M. (2024, May 23). *Elon Musk says AI will take all our jobs | CNN Business*. CNN. https://www.cnn.com/2024/05/23/tech/elon-musk-ai-your-job
Korn, J. (2023, April 12). *FBI warns consumers not to use public phone charging stations | CNN Business*. CNN. https://www.cnn.com/2023/04/12/tech/fbi-public-charging-port-warning
Lau, C. (2024, April 11). *Mexican drug gangs "increasingly targeting" Australia as meth supplies overtake rivals, police say*. CNN. https://www.cnn.com/2024/04/11/australia/australia-drugs-meth-mexican-cartels-intl-hnk
Madhok, D. (2024, May 16). *Microsoft asks some employees in China to move to other countries | CNN Business*. CNN. https://www.cnn.com/2024/05/16/tech/microsoft-china-employees-relocate-hnk-intl
Pfeifer, M. H. W., Hazel. (2024, March 22). *Onigiri Asakusa Yadoroku: Tokyo's oldest rice ball restaurant*. CNN. https://www.cnn.com/travel/tokyo-oldest-onigiri-rice-ball-shop-intl-hnk
Xu, X. (2024, March 25). *Paris waiters compete in race to get a coffee and croissant across the capital*. CNN. https://www.cnn.com/travel/paris-waiters-course-des-cafe-race
Ziady, H. (2024, May 16). *Facebook and Instagram probed over fears they may be too addictive for children*

| *CNN Business*. CNN. https://www.cnn.com/2024/05/16/tech/europe-investigation-meta-child-safety

References (Code for this assignment)

```
rm(list = ls())


library(slam)
library(tm)
library(SnowballC)
library(proxy)
library(igraph)


data_cname = file.path(".","Data_text")
```

Question 2

```
set.seed("31994695") data_docs = Corpus(DirSource(data_cname))


data_docs
```

Question 3

```
juneremoveChars ← content_transformer(function(x, pattern) gsub(pattern, "", x))
```

Tokenisation

```
data_docs ← tm_map(data_docs, removeNumbers) data_docs ← tm_map(data_docs, removePunctuation) data_docs ← tm_map(data_docs, content_transformer(tolower))

data_docs ← tm_map(data_docs, juneremoveChars, "’s")

data_docs ← tm_map(data_docs, juneremoveChars, "’ve")

data_docs ← tm_map(data_docs, juneremoveChars, "’d")
```

```
data_docs ← tm_map(data_docs, juneremoveChars, "’m")
data_docs ← tm_map(data_docs, juneremoveChars, "n't")
data_docs ← tm_map(data_docs, juneremoveChars, "like")
data_docs ← tm_map(data_docs, juneremoveChars, "can")
data_docs ← tm_map(data_docs, juneremoveChars, "take")
data_docs ← tm_map(data_docs, juneremoveChars, "also")
data_docs ← tm_map(data_docs, juneremoveChars, "one")
data_docs ← tm_map(data_docs, juneremoveChars, "year")
data_docs ← tm_map(data_docs, juneremoveChars, "–")
data_docs ← tm_map(data_docs, juneremoveChars, "“")
data_docs ← tm_map(data_docs, juneremoveChars, "”")
```

Remove stop words and white space

```
data_docs ← tm_map(data_docs, removeWords, stopwords("english")) data_docs
← tm_map(data_docs, stripWhitespace)
```

Stem

```
data_docs← tm_map(data_docs, stemDocument, language = "english")
```

Create document term matrix

```
set.seed("31994695") data_dtm ← DocumentTermMatrix(data_docs)
dim(data_dtm)
```

Remove sparse terms

Remove columns with 60% empty (0) cells

```
data_dtm ← removeSparseTerms(data_dtm, sparse = 0.5)
#write.csv(data_dtm, "data_dtm.csv")
dim(data_dtm)
```

Question 4

Cosine distance between each document for clustering.

```
set.seed("31994695")
dtms = as.matrix(data_dtm)
distmatrix = proxy::dist(dtms, method = "cosine")
fit = hclust(distmatrix, method = "ward.D")
plot(fit, hang = -1, main = "Question 4, Clustering Dendrogram")

#inspect(data_dtm)
fit ← hclust(distmatrix, method = "ward.D")
fit
```

using cluster object "fit" create required number of clusters.

```
cutfit ← cutree(fit, k = 3)
```

Calculate the accuracy with which the clustering groups documents by topic.

Create vector of topic labels in same order as corpus

```
topics = c("food", "food", "food","food","food", "tech", "tech", "tech", "tech",
"tech", "world", "world", "world", "world", "world")

groups = cutree(fit, k = 3)
cluster_table ← table(GroupNames = topics, Clusters = groups)
```

```
TA = as.data.frame.matrix(table(GroupNames = topics, Clusters = groups))
TA = TA[,c(2,1,3)]
TA

TA_matrix ← as.matrix(TA)
diag_TA ← diag(TA_matrix)
accuracy ← sum(diag_TA) / sum(TA_matrix)
accuracy
```

Question 5

convert to binary matrix

```
dtmsx = as.matrix((dtms > 0) + 0)
```

multiply binary matrix by its transpose

```
ByAbsMatrix = dtmsx %*% t(dtmsx)
```

make leading diagonal zero

```
diag(ByAbsMatrix) = 0
```

Create graph object

```
set.seed("31994695") Q5_SM_network =
graph_from_adjacency_matrix(ByAbsMatrix, mode = "undirected", weighted =
TRUE)
```

Plot the Basic model

```
set.seed("31994695") plot(Q5_SM_network, main = "Q5 Single Basic Mode
Network")
```

Get the weights

Q5_SM_network_weight = E(Q5_SM_network)$weight

Create color palette function

color_picker = colorRampPalette(c("pink","lightblue","green"))

Generate edge colors based on weights

Q5_edge_colors ← color_picker(length(Q5_SM_network_weight))
[as.numeric(cut(Q5_SM_network_weight, breaks =
length(Q5_SM_network_weight)))]

set.seed("31994695")

Plot the graph

plot(Q5_SM_network, edge.label = Q5_SM_network_weight, edge.color =
Q5_edge_colors, edge.width = 1, main = "Q5 Single Final Mode Network")

Question 6

set.seed("31994695") ByTokenMatrix = t(dtmsx) %*% dtmsx

make leading diagonal zero

diag(ByTokenMatrix) = 0

Create graph object

set.seed("31994695") Q6_TK_network =
graph_from_adjacency_matrix(ByTokenMatrix, mode = "undirected", weighted =
TRUE)

plot the basic model

set.seed("31994695") plot(Q6_TK_network, main = "Q6 Single Basic Mode Network")

Get the weights

Q6_TK_network_weight = E(Q6_TK_network)$weight

Generate edge colors based on weights

Q6_edge_colors ← color_picker(length(Q6_TK_network_weight)) [as.numeric(cut(Q6_TK_network_weight, breaks = length(Q6_TK_network_weight)))]

set.seed("31994695")

Plot the graph

plot(Q6_TK_network, edge.label = Q6_TK_network_weight, edge.color = Q6_edge_colors, edge.width = 1, main = "Q6 Token Final Mode Network")

Question 7

start with documnet term matrix dtms

dtmsa = as.data.frame(dtms) # clone dtms

dtmsa$ABS = rownames(dtmsa) # add row names

dtmsb = data.frame() for (i in 1:nrow(dtmsa)){ for (j in 1:(ncol(dtmsa) - 1)){ touse = cbind(dtmsa[i,j], dtmsa[i,ncol(dtmsa)],colnames(dtmsa[j])) dtmsb = rbind(dtmsb,touse)}} # close loops

colnames(dtmsb) = c("weight", "abs", "token")

dtmsc = dtmsb[dtmsb$weight != 0,] #delete 0 weights

put columns in order : abs, token, weight

dtmsc = dtmsc[,c(2,3,1)]

create graph object and declare bipartite

g ← graph.data.frame(dtmsc, directed = FALSE)

bipartite.mapping(g)

g set.seed("31994695") V(g)$type ← bipartite_mapping(g)$type V(g)$color ← ifelse(V(g)$type, "lightgreen","pink") V(g)$shape ← ifelse(V(g)$type, "circle", "square") E(g)$color ← "lightgrey"

plot the basic Bipartite network plot

#plot(g)

Get the weights

Q7_BP_network_weight = E(g)$weight

change to numeric

Q7_BP_network_weight ← as.numeric(Q7_BP_network_weight)

set.seed("31994695")

Generate edge colors based on weights

Q7_edge_colors ← color_picker(length(Q7_BP_network_weight))
[as.numeric(cut(Q7_BP_network_weight, breaks =
length(Q7_BP_network_weight)))]

Plot the graph

set.seed("31994695") plot(g, edge.label = Q7_BP_network_weight, edge.color = Q7_edge_colors, edge.width = 1, main = "Q7 Bipartite Final Mode Network")