FIT3152 – Data Analytics

MONASH University

NLP and Network Analysis

A natural language processing and network study of blogs on food, home, lifestyle, and travel

Joanna Moy

2. Converting Each Document to Text Format

After collecting a set of fifteen machine-readable text documents from various sites, we will create a corpus by first converting each document into a text format. To convert each blog post into a text format, I navigated to the blog post in my web browser and copied the main text content along with the title. I then opened a text editor (i.e. Notepad) and pasted the copied text into a new document, ensuring the title was at the top, followed by a newline and then the main content. I saved each document as a '.txt' file, using filenames that reflect the document's topic and source for easy identification.

3. Converting Each Document to Text Format

In preparing the corpus for analysis, several text transformations were performed to make sure that the data was clean, standardised, and focused on meaningful content. Firstly, all text was converted to lowercase to standardise the data and make the analysis case insensitive. This was achieved using the **'content_transformer(tolower)'** function in R. Numbers were then removed from the text as they generally do not contribute to the connotation in textual analysis and can create noise in the data. This was done using the **'removeNumbers'** function. Punctuation was also removed to eliminate unnecessary tokens that do not contribute to the meaning of words. This was performed using the **'removePunctuation'** function. Commonly used words, also known as stop words, were filtered out to focus on more meaningful terms and was accomplished using the **'removeWords'** function. Extra whitespaces, which might have been introduced during text cleaning, were stripped to ensure consistency, and was done using the **'stripWhitespace'** function. Lastly, words were stemmed to their root form to group similar terms together and reduce the complexity.

After preprocessing, the Document-Term Matrix (DTM) was created to quantify the presence of terms in each document. The final DTM contains approximately 20 tokens, providing a concise and relevant representation of the text data. The DTM has been included in the appendix as a table for reference (see Appendix A).

4. Hierarchical Clustering of the Corpus

The hierarchical clustering and resulting dendrogram, alongside the confusion matrix, provide an understanding into the quality of the clustering for the given corpus. This dendrogram will be shown in the appendix (see Appendix B).

The dendrogram illustrates the hierarchical relationships among the documents, but some clusters show significant overlap, indicating that the clustering algorithm had difficulty distinctly separating the documents based on their topics.

```
Confusion Matrix and Statistics

          Reference
Prediction  Food Home Lifestyle Travel
  Food         1    0         0      0
  Home         1    1         2      1
  Lifestyle    1    0         1      3
  Travel       0    1         2      1

Overall Statistics

               Accuracy : 0.2667
                 95% CI : (0.0779, 0.551)
    No Information Rate : 0.3333
    P-Value [Acc > NIR] : 0.7908

                  Kappa : 0.012

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Food Class: Home Class: Lifestyle
Sensitivity              0.33333     0.50000          0.20000
Specificity              1.00000     0.69231          0.60000
Pos Pred Value           1.00000     0.20000          0.20000
Neg Pred Value           0.85714     0.90000          0.60000
Prevalence               0.20000     0.13333          0.33333
Detection Rate           0.06667     0.06667          0.06667
Detection Prevalence     0.06667     0.33333          0.33333
Balanced Accuracy        0.66667     0.59615          0.40000
                     Class: Travel
Sensitivity                0.20000
Specificity                0.70000
Pos Pred Value             0.25000
Neg Pred Value             0.63636
Prevalence                 0.33333
Detection Rate             0.06667
Detection Prevalence       0.26667
Balanced Accuracy          0.45000
```

The clustering accuracy is 26.67% which is lower than the no-information rate of 33.33% suggesting that the clustering did not perform better than random guessing. The kappa statistic is very low at 0.012, indicating almost no agreement between the clustering results and the true labels.

In regards to class-wise performance, the sensitivity varies:

- Food: The sensitivity is 33.33%, indicating that only one-third of the food documents were correctly identified. The specificity is high (100%), but this could be due to the low prevalence of this class.
- Home: The sensitivity is 50%, showing moderate identification capability, with a specificity of 69.23%.
- Lifestyle: The sensitivity is 20%, reflecting poor identification, with a specificity of 60%.
- Travel: The sensitivity is 20%, also indicating poor performance, with a specificity of 70%.

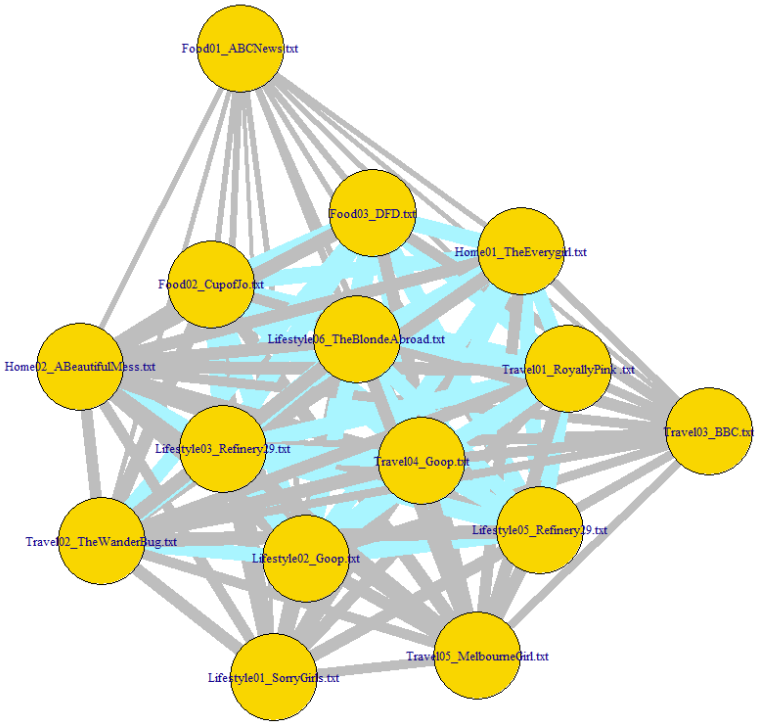
The mixed clusters in the dendrogram and the confusion matrix suggest that the documents from different categories share common words and themes, leading to confusion in clustering. This issue might be due to the removal of sparse terms, which could have resulted in the loss of discriminative features.

The clustering quality has been deemed suboptimal at best. The dendrogram shows that some documents from different topics are place closer together, indicating overlapping content or insufficient differentiation based on the selected features. Potential improvements could

involve adding more discriminative features, using different preprocessing techniques or experimenting with different clustering algorithms and parameter tuning.

5. Single-Mode Network Showing Connections Between Documents

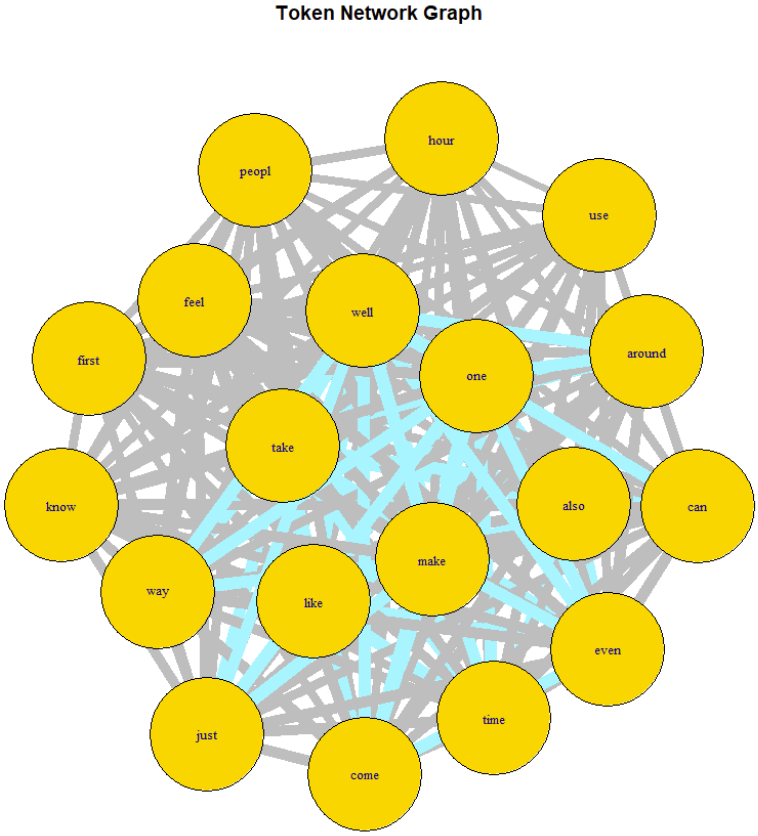
To calculate the connections between each document, a DTM was created, representing the frequency of terms in each document. This matrix was then binarized, converting all non-zero entries to 1, indicating the presence of a term. The connections between documents were calculated by multiplying the binary DTM by its transpose, resulting in a matrix where each element represents the number of shared terms between two documents. The diagonal was set to zero to exclude self-connections, creating an adjacency matrix used to construct the graph.



The network graph reveals the relationships between the documents based on shared terms. Documents with more connections are positioned centrally, indicating they share many terms with other documents. The graph shows several clusters, suggesting groups of documents that are thematically similar. For instance, the nodes with higher degrees, like those in the centre of the graph, indicate documents that are more central and share terms with many others. These central documents are crucial for understanding the core themes within the corpus.

The graph was improved by highlighting key features to improve readability and insight. Nodes were coloured based on their degree, with red indicating nodes with a higher number of connections and gold indicating fewer connections. The size of each node was scaled according to its degree, making more connected nodes larger. Edges were coloured based on their weight, with stronger connections coloured in blue and weaker connections in grey.

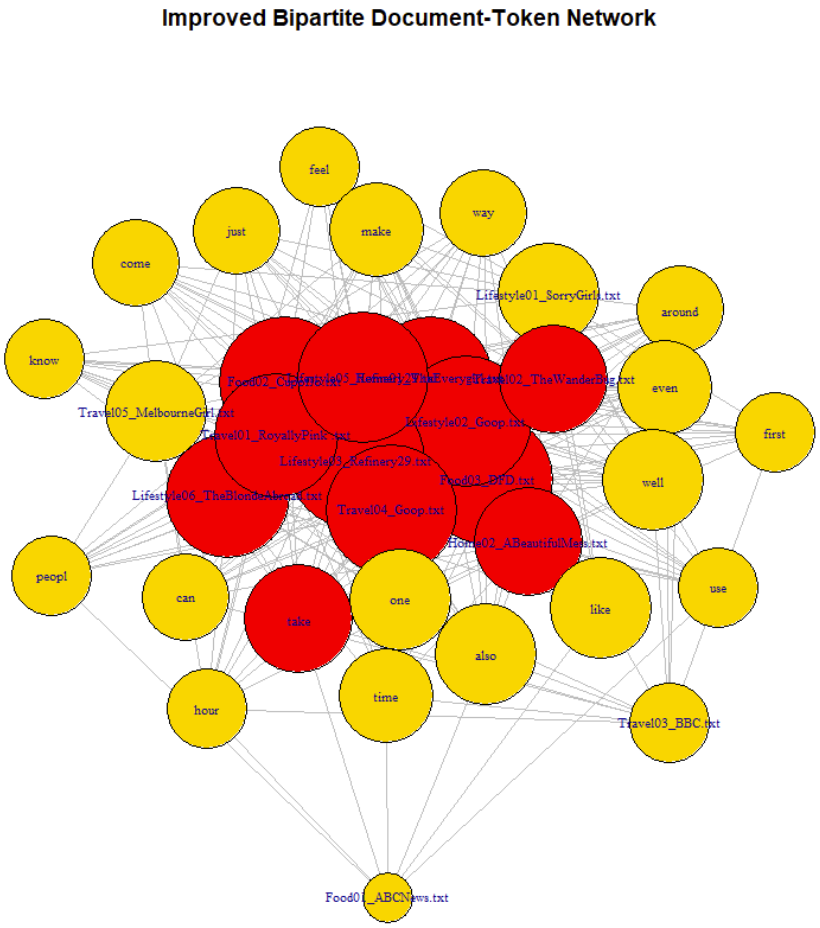
6. Single-Mode Network Showing Connections Between Tokens



For this question, similar steps to Question 5 were taken to calculate the connections between each token and improvements to highlight interesting features. As for the token network graph, it reveals the relationships between words based on their co-occurrence in the documents. Words with more connections are positioned centrally, indicating they co-occur with many other words across the documents. For example, words like “make”, “take”, and “one” are central, suggesting that they are commonly used in various contexts within the documents. Similarly, the edges in the graph represent the strength of the connections between words, with thicker and blue edges indicating stronger connections, which highlights which word pairs are most frequently used together.

7. Bipartite (Two-Mode) Network of the Corpus

To transform the data into a suitable format for a bipartite network, the DTM was first created, using similar methods to previous questions. This DTM will represent the frequency of terms in each document. The matrix was then binarized, converting all non-zero entries to 1, indicating the presence of a term. An edge list was also generated from the binary matrix, with each row representing an edge between a document and a token. This edge list was then used to create a bipartite graph, with node types set to differentiate between documents and tokens.



The following bipartite network graph illustrates the relationships between words and documents. The words (tokens) are connected to the documents in which they appear, revealing how terms are distributed across the documents. The central tokens are those that appear in multiple documents, indicating their common usage in the corpus. These central tokens were revealed to be words such as “one” and “take”. The graph also shows clusters of documents connected by common tokens, suggesting some sort of contextual similarities. For example, documents connected by specific tokens may belong to the same topic or discuss similar subjects, revealing clear groups within the data.

To improve the graph and highlight key features, several visual enhancements were made similar to previous questions likewise.

8. Summaries of Results

The bipartite network analysis revealed key documents and tokens within the corpus. Central tokens, such as “take,” “make,” and “one,” are common across multiple documents, indicating their importance in the text. Central documents, such as “Food02_CupofJo.txt” and “Travel05_MelbourneGirl.txt,” are connected to many tokens, suggesting they cover diverse topics. Clusters of documents connected by shared tokens suggest similarities, with clear groups forming within the data.

Clustering techniques group documents based on their content similarity, helping identify broader themed categories. However, they may overlook nuanced relationships between specific terms and documents. Social network analysis, on the other hand, highlights the detailed connections between words and documents, revealing specific terms’ centrality and document interconnections. While clustering provides an overview of thematic groups, social network analysis offers a granular view term-document relationships, making both methods complementary.

To better discriminate between the documents, we can implement several Natural Language Processing techniques. The first method involves Term Frequency-Inverse Document Frequency (TF-IDF) which weighs terms based on their frequency in a document relative to their frequency across all documents, helping highlight unique terms and reduce the impact of common words. Another method is to use Named Entity Recognition (NER), in which identify and categorise entities, such as names, locations, or dates, to better understand the context and specific subjects within the documents. We can also use topic modelling using Latent Dirichlet Allocation (LDA) to discover underlying topics in the corpus, providing a more structured representation of document themes.

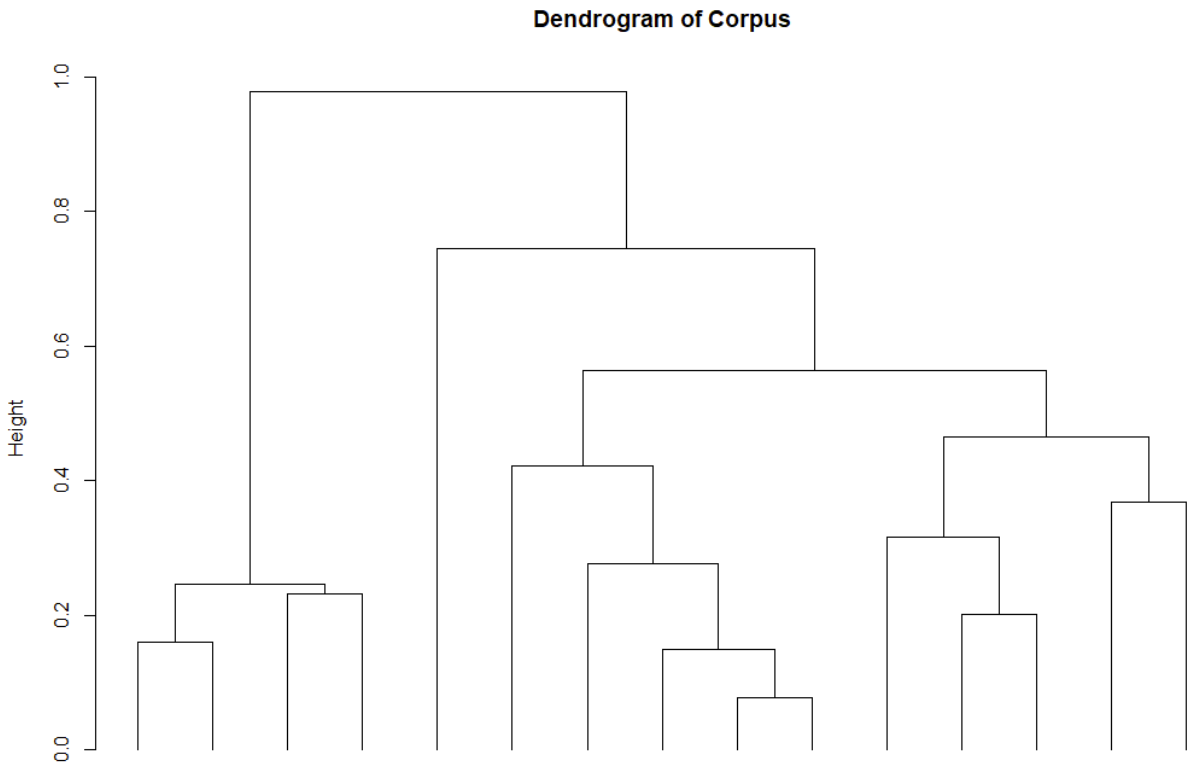
APPENDIX A

Document-Term Matrix (DTM)

	also	hour	like	people	take	time	use	around	can	come	even	feel	first	just	know	make	one	way	well
Food01_ABCNews.txt	1	1	1	3	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Food02_CupofJo.txt	4	1	11	1	5	7	0	3	4	1	1	8	3	5	2	6	11	3	3
Food03_DFD.txt	4	10	3	2	3	1	5	1	2	1	4	2	2	5	0	1	6	2	2
Home01_TheEverygirl.txt	10	0	3	4	3	6	2	0	3	1	1	6	1	3	3	7	10	3	3
Home02_ABeautifulMess.txt	4	1	2	0	2	7	20	1	17	0	0	0	2	3	3	3	3	2	2
Lifestyle01_SorryGirls.txt	1	0	1	0	1	0	1	1	0	2	1	1	2	1	1	1	1	0	1
Lifestyle02_Goop.txt	7	0	3	1	2	2	10	1	30	1	5	19	4	3	1	8	2	6	1
Lifestyle03_Refinery29.txt	16	4	11	2	8	6	5	1	9	2	4	8	5	3	4	9	15	6	1
Lifestyle05_Refinery29.txt	0	3	4	10	3	5	2	2	6	6	3	1	2	6	4	3	6	7	3
Lifestyle06_TheBlondeAbroad.txt	7	1	10	2	2	9	13	0	33	9	4	2	0	4	2	12	10	6	1
Travel01_RoyallyPink .txt	6	3	8	1	5	6	0	8	0	1	1	5	3	2	3	1	4	5	2
Travel02_TheWanderBug.txt	1	0	3	0	2	3	2	1	5	2	1	1	0	1	0	2	8	1	1
Travel03_BBC.txt	1	2	2	0	4	2	0	2	3	0	1	0	1	0	0	0	2	0	2
Travel04_Goop.txt	5	6	4	1	3	15	5	2	18	1	3	2	2	0	1	2	2	1	2
Travel05_MelbourneGirl.txt	2	1	0	2	1	0	0	2	6	1	2	0	0	1	2	2	6	3	3

APPENDIX B

Dendrogram of Resulting Hierarchical Clustering



APPENDIX C

All Tasks Implemented in R-Code

```
rm(list = ls())


# Question 3

library(slam)

library(tm)

library(SnowballC)


cname <- "C:\\Users\\Joanna
Moy\\Desktop\\Y3S1\\FIT3152\\Assessments\\Assignment
3\\Corpus\\txt"

print(dir(cname))


# Load documents into a corpus

docs <- Corpus(DirSource(cname))

print(summary(docs))


# Text transformations

docs <- tm_map(docs, content_transformer(tolower)) # Convert
to lowercase

docs <- tm_map(docs, removeNumbers) # Remove numbers

docs <- tm_map(docs, removePunctuation) # Remove punctuation

docs <- tm_map(docs, removeWords, stopwords("english")) #
Remove stop words

docs <- tm_map(docs, stripWhitespace) # Remove extra
whitespace

docs <- tm_map(docs, stemDocument, language = "english") #
Stem words


# Creates the DTM

dtm <- DocumentTermMatrix(docs)
```

```
# Remove sparse terms to get approximately 20 tokens
dtm <- removeSparseTerms(dtm, 0.33)

# Convert DTM to a data frame
dtm_df <- as.data.frame(as.matrix(dtm))

# Save the DTM to a CSV file
write.csv(dtm_df, "Document-Term Matrix.csv")

print(dtm_df)

ncol(dtm_df)



# Question 4
library(tm)
library(SnowballC)
library(cluster)
library(proxy)
library(caret)

# Convert DTM to a matrix
dtm_matrix <- as.matrix(dtm)

# Compute cosine distance
cosine_dist <- proxy::dist(dtm_matrix, method = "cosine")

# Perform hierarchical clustering
hclust_res <- hclust(cosine_dist, method = "ward.D2")
```

```
# Plot the dendrogram
plot(hclust_res, hang = -1, labels = FALSE, main = "Dendrogram
of Corpus")


# Identify clusters
clusters <- cutree(hclust_res, k = 4)
print(clusters)


# Assign cluster labels
cluster_labels <- as.factor(clusters)
levels(cluster_labels) <- c("Food", "Home", "Lifestyle",
"Travel")
print(cluster_labels)


# True labels for your documents
true_labels <- c("Food", "Food", "Food",
                 "Home", "Home",
                 "Lifestyle", "Lifestyle", "Lifestyle",
"Lifestyle", "Lifestyle",
                 "Travel", "Travel", "Travel", "Travel",
"Travel")


# Convert true labels to a factor
true_labels <- factor(true_labels, levels = c("Food", "Home",
"Lifestyle", "Travel"))


# Ensure the levels of clusters and true labels are the same
cluster_labels <- factor(cluster_labels, levels =
levels(true_labels))


# Print cluster_labels and true_labels to verify
print(cluster_labels)
```

```
print(true_labels)

# Compare clusters to true labels
confusionMatrix(cluster_labels, true_labels)




# Question 5
library(igraph)

# Convert DTM to binary matrix
dtmsx <- as.matrix(dtm)
dtmsx <- (dtmsx > 0) + 0

# Calculate connections between documents
ByAbsMatrix <- dtmsx %*% t(dtmsx)

# Make leading diagonal zero
diag(ByAbsMatrix) <- 0

# Create graph object
ByAbs <- graph_from_adjacency_matrix(ByAbsMatrix, mode =
"undirected", weighted = TRUE)

# Plot the graph
plot(ByAbs, vertex.size = 10, vertex.label.cex = 0.8,
edge.width = E(ByAbs)$weight)

# Improve visualization
# Highlight nodes based on degree
V(ByAbs)$color <- ifelse(degree(ByAbs) >
median(degree(ByAbs)), "red", "gold")
```

```
# Scale node size based on degree
V(ByAbs)$size <- degree(ByAbs) * 2

# Highlight edges based on weight
E(ByAbs)$color <- ifelse(E(ByAbs)$weight >
median(E(ByAbs)$weight), "cadetblue1", "gray")

# Plot the improved graph
plot(ByAbs, vertex.size = V(ByAbs)$size, vertex.label.cex =
0.8, vertex.color = V(ByAbs)$color, edge.width =
E(ByAbs)$weight, edge.color = E(ByAbs)$color)



# Question 6
# Convert DTM to binary matrix
dtmsx <- as.matrix(dtm)
dtmsx <- (dtmsx > 0) + 0

# Calculate connections between words
ByTokenMatrix <- t(dtmsx) %*% dtmsx

# Make leading diagonal zero
diag(ByTokenMatrix) <- 0

# Create graph object
ByToken <- graph_from_adjacency_matrix(ByTokenMatrix, mode =
"undirected", weighted = TRUE)

# Plot the graph
```

```
plot(ByToken, vertex.size = 10, vertex.label.cex = 0.8,
edge.width = E(ByToken)$weight)


# Improve visualization

# Highlight nodes based on degree

V(ByToken)$color <- ifelse(degree(ByToken) >
median(degree(ByToken)), "red", "gold")


# Scale node size based on degree

V(ByToken)$size <- degree(ByToken) * 2


# Highlight edges based on weight

E(ByToken)$color <- ifelse(E(ByToken)$weight >
median(E(ByToken)$weight), "cadetblue1", "gray")


# Plot the improved graph

plot(ByToken, vertex.size = V(ByToken)$size, vertex.label.cex
= 0.8, vertex.color = V(ByToken)$color, edge.width =
E(ByToken)$weight, edge.color = E(ByToken)$color, main =
"Token Network Graph")




# Question 7

# Create bipartite graph

# Create a data frame with document IDs and tokens

doc_ids <- rownames(dtmsx)

tokens <- colnames(dtmsx)

edges <- which(dtmsx == 1, arr.ind = TRUE)

edge_list <- data.frame(doc = doc_ids[edges[, 1]], token =
tokens[edges[, 2]])


# Create graph from edge list
```

```
bipartite_graph <- graph_from_data_frame(edge_list, directed =
FALSE)


# Set node types: TRUE for documents, FALSE for tokens

V(bipartite_graph)$type <- V(bipartite_graph)$name %in%
doc_ids


# Plot the bipartite graph

plot(bipartite_graph,

     vertex.label.cex = 0.7,

     vertex.color = ifelse(V(bipartite_graph)$type, "skyblue",
"salmon"),

     main = "Bipartite Document-Token Network")


# Improve visualization

# Highlight nodes based on degree

V(bipartite_graph)$color <- ifelse(degree(bipartite_graph) >
median(degree(bipartite_graph)), "red", "gold")


# Scale node size based on degree

V(bipartite_graph)$size <- degree(bipartite_graph) * 2


# Highlight edges based on weight (if applicable)

E(bipartite_graph)$color <- "gray"


# Plot the improved graph

plot(bipartite_graph,

     vertex.size = V(bipartite_graph)$size,

     vertex.label.cex = 0.7,

     vertex.color = V(bipartite_graph)$color,

     edge.width = 1,

     edge.color = E(bipartite_graph)$color,
```

```
main = "Improved Bipartite Document-Token Network")
```

References

Ang, D. (2023, November 30). 10 best Melbourne cafes & coffee shops – the top cafes you need to experience in the “Coffee Capital of Australia”. DanielFoodDiary.com. https://danielfooddiary.com/2023/11/30/melbournecafes/

Brown, L. (2024, June 1). Turkey’s new Mesopotamia Express takes visitors through the country’s rich history. BBC. https://www.bbc.com/travel/article/20240531-turkeys-new-mesopotamia-express-train

Collie, E. [Melbourne Girl]. (2022). The best markets in Melbourne to eat, shop, and explore. Melbourne Girl. https://www.melbournegirl.com.au/2018/11/07/the-best-markets-in-melbourne-to-eat-shop-and-explore/

Genevieve. (2024, March 17). Montmartre guide: The best things to do in Paris’ 18th Arrondissement. The Wander Bug. https://thewanderbug.com/montmartre-neighbourhood-guide-paris/

Harrison, O. (2021, July 6). Our out-of-office email situation is out of control. Refinery29. https://www.refinery29.com/en-us/2021/07/10557603/out-of-office-vacation-email-anxiety

John, D. (2024, May 30). What really works for jet lag? Goop. https://goop.com/wellness/health/what-really-works-for-jet-lag/

Larson, E. (2024, March 7). The best paint for kitchen cabinets. A Beautiful Mess. https://abeautifulmess.com/whats-the-best-paint-for-kitchen-cabinets/

Lyon, S. (2023, August 18). This 425-square-foot Newport Beach rental is a coastal grandma’s dream. The Everygirl. https://theeverygirl.com/sarah-horton-home-tour/

MacDermaid, K. (n.d.). Throwing a outdoor garden tea party *on a budget*. The Sorry Girls. https://www.thesorrygirls.com/lifestyle/cxqz6i9bwq08kqu7oifmx089gdsc71

McCarthy, K. (2024, June 7). Cream cheese recall expanded: Tillamook recalls cheese slices sold at Costco. ABC News. https://abcnews.go.com/GMA/Food/cream-cheese-recall-expanded-tillamook-recalls-cheese-slices/story?id=110922413

McGowan, E. (2024, May 22). When is humor a healthy coping mechanism — and when is it not? The Good Trade. https://www.thegoodtrade.com/features/humor-coping-mechanism/

Rosenstrach, J. (2022, August 3). Where New Yorkers ear in New York. Cup of Jo. https://cupofjo.com/2022/08/03/where-new-yorkers-eat-in-new-york/

Royally Pink. (n.d.). What I did during my first trip to London. Royally Pink. https://www.royallypink.com/2024/05/what-i-did-during-my-first-trip-to.html

Slinko, R. (2024, May 16). A week in the life of a 22-year-old Mary Kay consultant. Refinery29. https://www.refinery29.com/en-us/week-in-the-life-mary-kay-consultant

The Blonde Abroad. (n.d.). Easy ways to use alternative to plastic. The Blonde Abroad. https://www.theblondeabroad.com/easy-ways-to-use-alternatives-to-plastic/