FIT 3152
Assignment 03
33370311

Q1)
The machine readable text I have chosen comprise over the topics of Health, Movie Reviews and Technology articles just to give an overview our health1-health5 documents contain these informations respectively:
Research on semaglutide showing its efficacy in reducing cardiovascular risks in diabetes and kidney disease patients.
Study on antinephrin autoantibodies in glomerular diseases, suggesting their role in disease activity.
Analysis of global elective healthcare quality using inguinal hernia repairs as a model to assess access across different income levels.
Overview of the global HIV/AIDS epidemic, focusing on management, treatment accessibility, and challenges.
WHO report advocating for a shift from long-stay mental health institutions to community-based care in Southeast Asia to improve quality of life and reduce stigma

Then our movie documents contain these information from review1 - review4 respectively:
"Young Woman and the Sea" depicts Trudy Ederle's historic English Channel swim, blending inspiration with historical liberties.
"John Wick" revitalizes action cinema with its choreography and Keanu Reeves's performance.
"Furiosa" explores the backstory of a "Mad Max" character with expansive storytelling and action.
"Oppenheimer" delves into the ethical and moral dilemmas faced by J. Robert Oppenheimer during the development of the atomic bomb.

Laslty our technology documents from tech1-tech6 contain these information respectively:
Discussion on Mark Zuckerberg’s push for open-source AI at Meta, contrasting with other tech giants.
The use of AI-enhanced bionics helping Ukrainian soldiers regain functionality after injuries.
Dynamics between BYD and Tesla in the EV market, highlighting BYD’s competitive strategies.
Changes in Google's search algorithm affecting content visibility and raising content creators' concerns.
Survey showing low daily usage rates of generative AI tools despite significant investment.
Debate over privacy concerns between WhatsApp and Elon Musk regarding data handling practices.

Q2)
Most of the text I found was from online articles, hence I used the text in those article to create a text file for each text file a name based on its topic and id for example health5 where health is our topic and 5 is the id of that text document. Consequently, when I had to create the corpus I achieved this using the Corpus() function from the tm package in R, which is designed to handle collections of text documents. We specified DirSource(path_to_files, encoding = "UTF-8") as the input to this function, where DirSource() directs the function to our folder containing the text files. This folder

path is predefined in the path_to_files variable, which has our targeted textual data. The specification of UTF-8 encoding ensures that the text is correctly interpreted. By executing this command, we successfully transform loose, unstructured text files into a structured corpus format.

Q3)
For the preprocessing phase of the text analysis, several techniques were meticulously applied to refine the corpus for subsequent analytical tasks. Initially, the text was converted to lowercase to standardize the format across all documents, eliminating any discrepancies that case differences might introduce. This was followed by removing all punctuation and numeric characters to focus purely on textual content. Stopwords, which are commonly used words that offer little value in terms of text analysis (such as "and", "the", "is"), were also removed to reduce noise and focus on more meaningful words. Additionally, whitespace was stripped to clean up any extra spaces, and stemming was applied using the SnowballC library to reduce words to their base or root form, allowing for the consolidation of different forms of a word into a single representative form. These preprocessing steps were crucial in simplifying the text data, enabling more effective and efficient analysis in subsequent stages. Two custom functions were also employed because upon inspection some documents still had punctuations which had to be removed hence had to implement these custom functions one to further clean punctuation and another to remove URLs since most of my documents had URL in them, ensuring that the text was devoid of extraneous elements that could skew the analysis.

Following the initial text preprocessing steps, we proceeded to construct the main Document-Term Matrix (DTM) using the DocumentTermMatrix() function from the tm package. The DTM is a critical component in text mining as it converts the corpus into a matrix format where each row represents a document and each column represents a term or word from the corpus. The values in the matrix indicate the frequency of each term in each document.

Given the extensive preprocessing, the DTM initially included a large number of terms, many of which were sparse across the documents. To optimize the DTM for analysis, we applied the removeSparseTerms() function with a sparsity threshold of 0.45. This threshold was chosen after several iterations to balance between reducing sparsity and retaining enough meaningful terms for analysis. A lower sparsity threshold tends to retain more terms which might not be significant across the documents, whereas a higher threshold might exclude potentially insightful terms.

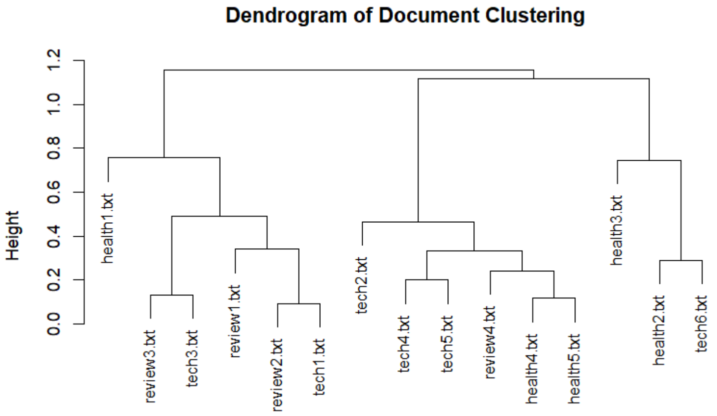
The resulting sparse DTM contained 12 terms across 15 documents, with a sparsity level of 32%. This dimensionality was deemed appropriate as it allowed for a manageable yet informative set of terms for clustering and network analysis, without overwhelming the models with noise or irrelevant data.

The inspection of the sparse DTM provided insight into the distribution and frequency of terms across different documents. For example, terms like "also", "can", and "people" appeared frequently across several documents, suggesting these words were significant within the context of the corpus. This observation was instrumental in understanding the overarching themes and discussions within the collected documents.

However, it's important to note that while the optimal number of terms aimed for was around 20, the actual number retained was 12. This decision was based on empirical observations where increasing the number of terms (by adjusting the sparsity threshold) led to lower clustering accuracy. The clustering models performed better with a slightly reduced set of terms, indicating a more focused and relevant set of data points for grouping the documents.

You can find our dtm in the appendix.

Q4.)



The dendrogram generated from hierarchical clustering using the cosine distance and Ward's method provides a visual representation of the relationships and similarities between documents based on their textual content. Each branch of the dendrogram represents a document, and the height at which branches merge represents the distance or dissimilarity between clusters.

Cluster Composition and Analysis:

- **Cluster 1** predominantly consists of movie review documents (review1.txt, review2.txt, review3.txt) and a mixture of technology documents (tech1.txt, tech3.txt). This cluster formation suggests that the language or thematic elements used in these movie reviews might share some vocabulary or stylistic features with the technology articles, perhaps due to discussions of technology in movies or the analytical nature of the reviews.
- **Cluster 2** captures mostly health documents (health2.txt, health3.txt) and a technology document (tech6.txt). The inclusion of a technology document in this cluster could indicate the use of specific medical or scientific terminology that overlaps with technology discussions, particularly if the technology document discusses health-related technologies.
- **Cluster 3** contains a mix of health (health1.txt, health4.txt, health5.txt), technology (tech2.txt, tech4.txt, tech5.txt), and a movie review (review4.txt). This diverse grouping

might be explained by overlapping themes or shared technical language, such as statistical analysis or scientific methodologies, which are common in both technical and research-focused health documents. For example our tech2.txt contains information on AI-enhanced bionics helping Ukrainian soldiers regain functionality after injuries. This article though belongs to technology but discusses health related content too which is causing health documents to cluster in the same cluster.

Contingency Table and Accuracy: The contingency table for the clustering results indicates a mixed distribution of documents across the clusters

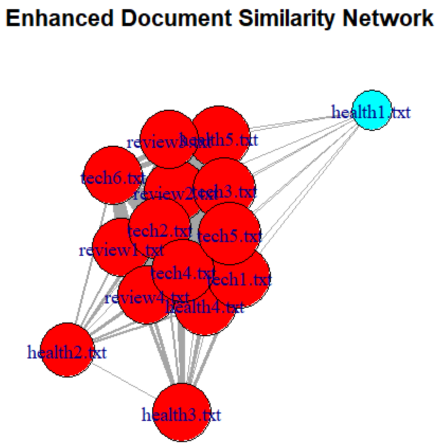
From this table, it's evident that there isn't a one-to-one correspondence between the actual categories and the clusters, reflecting some level of misclassification. For instance, while movie reviews mostly group together in Cluster 1, they also appear in Cluster 3. Similarly, technology and health documents are spread across all three clusters.

Accuracy Calculation: Given the mixed groupings, the accuracy of clustering is calculated as the ratio of correctly classified documents to the total number of documents. In this case:

Accuracy = (Correct classifications) / (Total documents) = (3 Movie Reviews in Cluster 1 + 2 Health in Cluster 2 + 3 Technology in Cluster 3) / 15 = 8 / 15 = 53.33%
This accuracy rate suggests moderate effectiveness in clustering, highlighting room for improvement in the preprocessing or clustering approach.

Analysis Conclusion: The clustering, while insightful, shows that the distinctions between some document types are not entirely clear-cut, possibly due to overlapping vocabulary or the multifaceted nature of the content. Improving the clustering accuracy might involve refining the preprocessing steps, such as enhancing the stopwords list or employing more nuanced text normalization techniques.

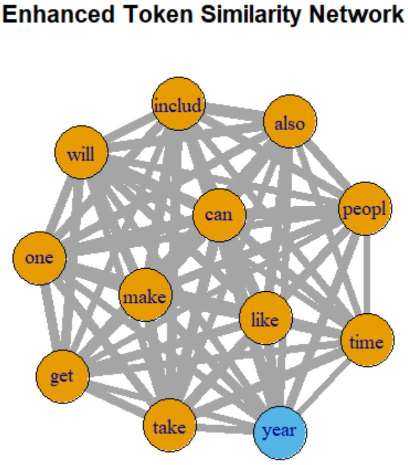
Q5.)



The network graph we have obtained in question 5 presents a clear picture of the connectivity and the themes of the various documents and it is seen to be significantly interconnected having a dense center mainly formed of technology and movie review documents. This core, where documents such as tech1. txt, tech3. txt, and review1. This is because if two txt are depicted as central nodes, it is likely that there is a high lexical and/or thematic similarity, probably as a result of the participation in discussions on themes technological in movies or technological focus in txt which are often contained in documents. Interestingly, it also outlines another less entangled system of health documents, namely health2. txt and health3. txt, which although are closely related to each other, are less integrated with the rest of the dense nucleus, signifying their marginal thematic profile, primarily focused on the topics of health that are not as intertwined with the technological concerns of the rest of the documents. Rather indicative is the fact that health1 is situated rather isolated. txt also indicates that it occupies a specific area within the health category and has different characteristics from the other documents of the corpus, and more so it may be recognized as an outlier or a specialist document. In summary we see that health documents tend to be different from technology and movie review documents.

Q6.)

The Token Similarity Network graph I've analyzed provides a comprehensive visualization of how key terms or tokens from my corpus are interconnected based on their co-occurrence across documents.



Network Structure and Term Relationships:

- In the network, nodes represent tokens and edges indicate the co-occurrence frequency between these tokens. Thicker edges suggest that two terms frequently appear together, highlighting their thematic or contextual relatedness. Central nodes such as "also," "can," and "people" are highly connected, suggesting their role as pivotal connectors or thematic hubs within the text, appearing frequently across various contexts.

Node Analysis:

- **Color Coding:** The nodes are color-coded, with most nodes in yellow except for "year," which is in blue. This distinction in color likely points to "year" having a unique contextual role separate from the other terms, possibly associated with specific temporal discussions within the corpus.
- **Size Variation:** All nodes appear similarly sized in this visualization, which typically might indicate uniform importance. However, scaling node sizes based on degree centrality could further highlight terms that hold more structural significance due to their numerous connections.

Interpretation of Key Terms:

- **High-Degree Nodes:** Terms like "also," "can," and "people" likely serve as generic connectors or bridging terms, facilitating diverse discussions across the corpus. Their extensive connectivity underscores their versatility and prevalence in linking different thematic areas.
- **Peripheral Nodes:** The term "year," shown in blue and somewhat peripheral, suggests it may be contextually specific, linked to discussions that involve temporal aspects, distinguishing it from more universally applicable terms.

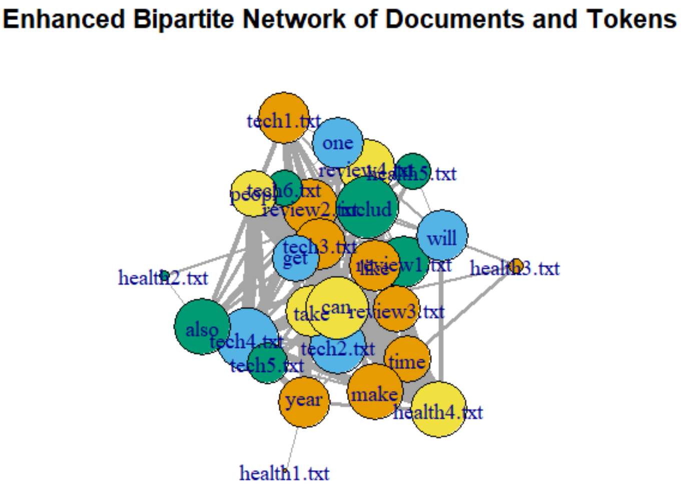
Cluster and Community Dynamics:

- The clustering pattern shows a dense grouping of terms, with "year" as a notable exception. This setup implies a strong thematic coherence among most terms, while "year" indicates specialized discussions that might not integrate as broadly with other topics.
- The network’s layout suggests that while many terms are broadly connected, reflecting a shared thematic framework, there are nuances, such as those highlighted by the outlier "year," which could signify specialized or less frequent discussions.

Strategic Insights for Text Analysis:

- **Community Detection:** I plan to delve deeper into community detection within the network to better understand the subtle groupings and perhaps uncover underlying sub-themes or specific contexts that certain terms, especially outliers like "year," are associated with. This will help refine my understanding of the corpus's structure and enhance the granularity of my text analysis.

Q7.) Analyzing the Enhanced Bipartite Network of Documents and Tokens, I've mapped out the connectivity between specific documents and tokens, reflecting the significance and thematic relevance of each token within different documents. Here's a breakdown of each token and its relationship with various documents, illustrating how these connections provide insights into the corpus's content:


Enhanced Bipartite Network of Documents and Tokens

Tokens:

1. **Include**: This token shows strong connectivity, particularly with review and technology documents. Its prevalence suggests discussions that often involve lists or specifications, common in technical descriptions or reviews.
2. **Will**: Frequently used in both predictive contexts and expressions of intent, "will" links across various document types, indicating a general utility in language usage across topics.
3. **People**: As a token, "people" appears primarily connected to health documents, hinting at discussions centered around patients, medical staff, or societal health impacts.
4. **Make**: This action-oriented term is versatile, found in both technical and review documents, possibly discussing how things are made (in tech) or plot developments (in reviews).
5. **Like**: Often used in comparisons or preferences, "like" is seen connecting with reviews where evaluative language is prevalent.
6. **Time**: Typically associated with discussions involving durations or sequences, "time" is crucial in documents discussing processes or historical contexts, visible in both health and review contexts.
7. **Year**: This temporally specific token is significant in documents that discuss events or results over time, such as annual reports or longitudinal studies in health documents.
8. **Take**: Found in instructional or procedural discourse, "take" connects strongly with health documents, likely in the context of medical guidelines or treatments.

9. **Get**: A general-purpose verb seen across various documents, "get" could relate to obtaining results, achieving goals, or literal acquisition, relevant in multiple contexts.
10. **Also**: A connector token, "also" links ideas and statements, prevalent across all document types, enhancing narrative flow and adding information.

Documents:

1. **Tech Documents (e.g., tech1.txt, tech2.txt, tech3.txt)**: These documents are heavily linked with tokens like "include," "make," and "will," which align with discussions on technological processes, features, and future developments.
2. **Health Documents (e.g., health1.txt, health2.txt, health3.txt)**: Health-related texts connect with "people," "take," and "year," indicating a focus on patient care, treatment regimens, and longitudinal health studies.
3. **Review Documents (e.g., review1.txt, review2.txt)**: Reviews tie strongly with evaluative tokens like "like" and process-oriented terms such as "make," reflecting the critical analysis and comparative discussions typical in reviews.

The connectivity patterns in this bipartite network not only underscore the specific uses of tokens within various thematic contexts but also highlight how certain terms serve as pivotal links, weaving through the corpus and binding different documents through shared language.

Appendix

References for Text Documents

Agency. (2024, May 23). *'Furiosa: A Mad Max Saga' review: Brilliant prequel that pulls hope from chaos*. The Star. Retrieved June 2, 2024, from https://www.thestar.com.my/lifestyle/entertainment/2024/05/23/039furiosa-a-mad-max-saga039-rview-brilliant-prequel-that-pulls-hope-from-chaos

BAHR, L. (2024, May 30). *Movie Review: Daisy Ridley shines in inspirational swimming pic 'Young Woman and the Sea'*. AP. Retrieved June 2, 2024, from https://apnews.com/article/young-woman-sea-movie-review-a238dd8f268ad1816a16e5c44aab9df4

Bergeron, R. (2024, May 22). *How AI and bionics are helping Ukrainian soldiers return to action*. CNN. Retrieved June 2, 2024, from https://edition.cnn.com/2024/05/22/tech/how-ai-and-bionics-are-helping-ukrainian-soldiers-return-to-action/index.html

Gerken, T. (2024, May 27). *WhatsApp boss in online spat with Elon Musk*. BBC. Retrieved June 2, 2024, from https://www.bbc.com/news/articles/c0ddwymz8ero

Germain, T. (2024, May 25). *Google just updated its algorithm. The Internet will never be the same*. BBC. Retrieved June 2, 2024, from https://www.bbc.com/future/article/20240524-how-googles-new-algorithm-will-shape-your-internet

Henge, F. E., Dehde, S., Lassé, M., Zahner, G., Seifert, L., & Schnarre, A. (2024, May 25). *Autoantibodies Targeting Nephrin in Podocytopathies*. PubMed. Retrieved June 2, 2024, from https://pubmed.ncbi.nlm.nih.gov/38804512/

Isaac, M. (2024, May 23). *How A.I. Made Mark Zuckerberg Popular Again in Silicon Valley*. New York Times. Retrieved June 2, 2024, from https://www.nytimes.com/2024/05/29/technology/mark-zuckerberg-meta-ai.html

Lemire, C. (2014, October 24). *John Wick*. Wikipedia. Retrieved June 2, 2024, from https://www.rogerebert.com/reviews/john-wick-2014

Morrow, A. (2024, May 30). *Elon Musk once mocked China's BYD. Now it's running circles around Tesla*. CNN. Retrieved June 2, 2024, from https://edition.cnn.com/2024/05/30/tech/elon-musk-byd-china-nightcap/index.html

NIHR Global Health Research Unit on Global Surgery. (2024, May 23). *Access to and quality of elective care: a prospective cohort study using hernia surgery as a tracer condition in 83 countries*. Wikipedia. Retrieved June 2, 2024, from https://pubmed.ncbi.nlm.nih.gov/38797188/

Perkovic, V., Tuttle, K. R., Rossing, P., & Mahaffey, K. W. (2024, May 24). *Effects of Semaglutide on Chronic Kidney Disease in Patients with Type 2 Diabetes*. PubMed. Retrieved June 2, 2024, from https://pubmed.ncbi.nlm.nih.gov/38785209/

Singleton, T. (2024, May 27). *AI products like ChatGPT much hyped but not much used, study says*. BBC. Retrieved June 2, 2024, from https://www.bbc.com/news/articles/c511x4g7x7jo

TECHERA, T. (2023, July 21). *Oppenheimer and the Last Great America*. Wikipedia. Retrieved June 2, 2024, from https://rlo.acton.org/archives/124695-oppenheimer-and-the-last-great-america.html?utm_term=oppenheimer%20movie&utm_campaign=&utm_source=adwords&utm_medium=ppc&hsa_acc=9098040689&hsa_cam=20398082377&hsa_grp=153243429804&hsa_ad=666737688398&hsa_src=g&hsa_tgt

Word Health Organisation. (n.d.). *Deinstitutionalize mental health care, strengthen community-based services: WHO*. Word Health Organisation. Retrieved June 2, 2024, from
https://www.who.int/southeastasia/news/detail/12-03-2024-deinstitutionalize-mental-health-care--
s
trengthen-community-based-services--who

World Health Organisation. (2023, July 13). *HIV and AIDS*. World Health Organisation. Retrieved June 2, 2024, from https://www.who.int/news-room/fact-sheets/detail/hiv-aids

DTM Table

Terms

Docs	year	also	includ	take	time	can	get	like	make	peopl	will	one
health1.txt	1	0	0	0	0	0	0	0	0	0	0	0
health2.txt	0	1	2	0	0	0	0	0	0	0	0	0
health3.txt	0	0	2	2	3	0	0	0	0	0	0	0
health4.txt	3	6	8	6	1	15	3	2	1	27	4	0
health5.txt	2	3	2	0	0	3	0	0	0	4	1	1
review1.txt	0	5	1	2	2	1	2	3	2	0	1	2
review2.txt	4	2	1	2	2	2	1	4	2	1	0	1

```
review3.txt  2  0   0  1  1  1  2  1  2   0  1  2
review4.txt  0  4   1  3  4  7  1  2  4   6  2  2
tech1.txt    6  4   2  3  4  4  0  2  4   1  0  1
tech2.txt    1  3   0  1  1  4  1  2  1   2  8  4
tech3.txt    4  0   1  1  0  1  1  1  4   1  2  2
tech4.txt    6  1   3  1  2  4  3  2  6  10  7  8
tech5.txt    1  1   3  0  0  1  0  2  2   6  3  0
tech6.txt    0  3   3  0  0  2  1  0  2   0  2  1
```

Code

```
> #Remove all objects from the workspace
> rm(list = ls())
> setwd("C:/Users/Home/OneDrive/Desktop/3152/Assignment 3") #-- OUR WORKING DIRECTORY
> #Remove all objects from the workspace
> rm(list = ls())
> #necessary libraries
> library(tm)         # Text mining
> library(SnowballC)   # Text stemming
> library(proxy)       # Similarity measures
> library(stats)       # Statistical functions
> library(igraph)      # Network analysis
> #path to the folder containing text files
> path_to_files <- "C:/Users/Home/OneDrive/Desktop/3152/Assignment 3/Text files"
> # Create the corpus
> docs <- Corpus(DirSource(path_to_files, encoding = "UTF-8"))
> # Extract content and metadata

> doc_content <- sapply(docs, as.character)

> doc_metadata <- data.frame(Document = names(doc_content), Content = doc_content)

> # Write the data to a CSV file

> write.csv(doc_metadata, "C:/Users/Home/OneDrive/Desktop/3152/Assignment
3/CorpusData.csv", row.names = FALSE)

> # Preprocess the text

> # Converting all text to lowercase

> docs <- tm_map(docs, content_transformer(tolower))

> # Removing punctuation

> docs <- tm_map(docs, removePunctuation)

> # Removing punctuation

> docs <- tm_map(docs, removePunctuation)

> # Removing numbers

> docs <- tm_map(docs, removeNumbers)

> # Stopwords removal

> docs <- tm_map(docs, removeWords, stopwords("english"))

> # Stripping whitespace

> docs <- tm_map(docs, stripWhitespace)

> # Stem documents

> docs <- tm_map(docs, stemDocument)

> # Custom punctuation removal

> removePunctuation2 <- content_transformer(function(x) gsub("[[:punct:]]+", " ", x))

> docs <- tm_map(docs, removePunctuation2)

> # Custom URL removal

> removeURL <- content_transformer(function(x) gsub("http[[:alnum:]]*", "", x))
> docs <- tm_map(docs, removeURL)

> # Creating the main Document-Term Matrix (DTM) for all documents

> dtm <- DocumentTermMatrix(docs)

> # Removing sparse terms to reduce the dimensionality of the DTM

> dtm_sparse <- removeSparseTerms(dtm, 0.45)

> dim(dtm_sparse)

[1] 15 12

> # Convert the sparse DTM to a matrix

> dtmsx <- as.matrix(dtm_sparse)

> # Inspect the updated sparse DTM

> inspect(dtm_sparse)

<<DocumentTermMatrix (documents: 15, terms: 12)>>

Non-/sparse entries: 123/57

Sparsity           : 32%

Maximal term length: 6

Weighting          : term frequency (tf)

Sample             :

             Terms

Docs          also can includ like make one peopl take will year

  health4.txt    6  15      8    2    1   0    27    6    4    3

  health5.txt    3   3      2    0    0   1     4    0    1    2

  review1.txt    5   1      1    3    2   2     0    2    1    0

  review2.txt    2   2      1    4    2   1     1    2    0    4

  review4.txt    4   7      1    2    4   2     6    3    2    0

  tech1.txt      4   4      2    2    4   1     1    3    0    6
```

```
 tech2.txt    3  4    0   2   1  4    2   1   8   1

 tech3.txt    0  1    1   1   4  2    1   1   2   4

 tech4.txt    1  4    3   2   6  8   10   1   7   6

 tech5.txt    1  1    3   2   2  0    6   0   3   1

> print(dtmsx)

          Terms

Docs         year also includ take time can get like make peopl will one

 health1.txt   1   0    0   0   0  0  0   0   0    0   0  0

 health2.txt   0   1    2   0   0  0  0   0   0    0   0  0

 health3.txt   0   0    2   2   3  0  0   0   0    0   0  0

 health4.txt   3   6    8   6   1 15  3   2   1   27   4  0

 health5.txt   2   3    2   0   0  3  0   0   0    4   1  1

 review1.txt   0   5    1   2   2  1  2   3   2    0   1  2

 review2.txt   4   2    1   2   2  2  1   4   2    1   0  1

 review3.txt   2   0    0   1   1  1  2   1   2    0   1  2

 review4.txt   0   4    1   3   4  7  1   2   4    6   2  2

 tech1.txt    6   4    2   3   4  4  0   2   4    1   0  1

 tech2.txt    1   3    0   1   1  4  1   2   1    2   8  4

 tech3.txt    4   0    1   1   0  1  1   1   4    1   2  2

 tech4.txt    6   1    3   1   2  4  3   2   6   10   7  8

 tech5.txt    1   1    3   0   0  1  0   2   2    6   3  0

 tech6.txt    0   3    3   0   0  2  1   0   2    0   2  1

> # Calculate the cosine distance between documents

> cosine_dist <- dist(dtmsx, method = "cosine")

> # Perform hierarchical clustering using the cosine distance
> hc <- hclust(cosine_dist, method = "ward.D2")

> # Plot the dendrogram

> plot(hc, main = "Dendrogram of Document Clustering", xlab = "", sub = "", cex = 0.9)

> # Identify clusters (let's assume we want 3 clusters for simplicity)

> clusters <- cutree(hc, k = 3)

> # Print cluster assignments

> print(clusters)

health1.txt health2.txt health3.txt health4.txt health5.txt review1.txt review2.txt review3.txt
review4.txt

          1           2           2           3           3           1           1           1           3

  tech1.txt   tech2.txt   tech3.txt   tech4.txt   tech5.txt   tech6.txt

          1           3           1           3           3           2

> # Save the cluster assignments for further analysis

> cluster_assignments <- data.frame(Document = rownames(dtmsx), Cluster = clusters)

> write.csv(cluster_assignments, file = "cluster_assignments.csv", row.names = FALSE)

> # Assuming you have the true labels for the topics of your documents

> true_labels <- c(rep("Health", 5), rep("Movie Review", 4), rep("Technology", 6))

> # Create a data frame with true labels and predicted clusters

> results <- data.frame(Document = rownames(dtmsx), TrueLabel = true_labels, Cluster = clusters)

> # Create a contingency table

> contingency_table <- table(results$TrueLabel, results$Cluster)

> print(contingency_table)



               1 2 3

  Health       1 2 2

  Movie Review 3 0 1
  Technology   2 1 3
> # Map clusters to true labels based on the contingency table
> cluster_to_label <- c("1" = "Movie Review", "2" = "Health", "3" = "Technology")
> results$MappedCluster <- factor(results$Cluster, levels = names(cluster_to_label), labels =
cluster_to_label)
> # Calculate accuracy
> accuracy <- sum(results$TrueLabel == results$MappedCluster) / nrow(results)
> print(paste("Clustering Accuracy:", accuracy))
[1] "Clustering Accuracy: 0.533333333333333"
> # Network Analysis
> # Convert the sparse DTM to a binary matrix (presence/absence of terms)
> dtmsx_binary <- as.matrix((dtmsx > 0) + 0)
> ByAbsMatrix <- dtmsx_binary %*% t(dtmsx_binary)
> diag(ByAbsMatrix) = 0
> # Convert the similarity matrix to a graph
> ByAbs <- graph_from_adjacency_matrix(ByAbsMatrix, mode = "undirected", weighted = TRUE)
> # Perform Fast Greedy Clustering
> fgc <- cluster_fast_greedy(ByAbs)
> # Assign community membership as vertex attribute
> V(ByAbs)$community <- membership(fgc)
> # Define colors for the communities
> colors <- rainbow(max(membership(fgc)))
> # Set vertex colors based on community membership
> V(ByAbs)$color <- colors[membership(fgc)]
> # Set vertex size based on degree centrality
> V(ByAbs)$size <- degree(ByAbs) * 3
> # Set edge width based on weight
> E(ByAbs)$width <- E(ByAbs)$weight
> # Plot the graph with enhanced features
> plot(ByAbs, vertex.label = V(ByAbs)$name, vertex.size = V(ByAbs)$size, vertex.color = V(ByAbs)$color, edge.width = E(ByAbs)$width, main = "Enhanced Document Similarity Network")
> # Create a binary matrix for term co-occurrence
> dtmsx = as.matrix(dtm_sparse)
> dtmsx = (dtmsx > 0) + 0
> ByTokenMatrix = t(dtmsx) %*% dtmsx
> diag(ByTokenMatrix) = 0
> # Convert the matrix to a graph
> ByToken = graph_from_adjacency_matrix(ByTokenMatrix, mode = "undirected", weighted = TRUE)
> # Filter edges by weight
> E(ByToken)$weight <- E(ByToken)$weight
> ByToken <- delete_edges(ByToken, E(ByToken)[weight < 2])
> # Detect communities
> communities <- cluster_fast_greedy(ByToken, weights = E(ByToken)$weight)
> # Color nodes by community
> V(ByToken)$color <- communities$membership
> # Adjust node size by degree
> V(ByToken)$size <- degree(ByToken) * 3
> # Plot the graph
> plot(ByToken, vertex.label = V(ByToken)$name, vertex.size = V(ByToken)$size, vertex.color = V(ByToken)$color, edge.width = E(ByToken)$weight, main = "Enhanced Token Similarity Network")
> # Bipartite Network Analysis
> # Generate the incidence matrix from the DTM
> incidence_matrix <- as.matrix(dtm_sparse)

> # Create the bipartite graph

> bipartite_graph <- graph.incidence(incidence_matrix)

> # Calculate edge weights based on term frequency

> edge_weights <- c()

> for (e in E(bipartite_graph)) {

+   ends <- ends(bipartite_graph, e)

+   from <- ends[1]

+   to <- ends[2]

+   edge_weights <- c(edge_weights, incidence_matrix[from, to])

+ }

> # Set edge weights

> E(bipartite_graph)$weight <- edge_weights

> # Detect communities in the bipartite network

> bipartite_communities <- cluster_fast_greedy(bipartite_graph, weights =
E(bipartite_graph)$weight)

> # Set vertex color by community

> V(bipartite_graph)$color <- bipartite_communities$membership

> # Adjust node size based on degree

> V(bipartite_graph)$size <- degree(bipartite_graph) * 3

> # Use a different layout algorithm to improve node separation (Kamada-Kawai layout)

> layout <- layout_with_kk(bipartite_graph, weights = E(bipartite_graph)$weight)

> # Plot the enhanced bipartite network with improved layout

> plot(bipartite_graph, vertex.label = V(bipartite_graph)$name, vertex.size = V(bipartite_graph)$size,

+      vertex.color = V(bipartite_graph)$color, edge.width = E(bipartite_graph)$weight,

+      layout = layout, main = "Enhanced Bipartite Network of Documents and Tokens")
```