

FIT 3152 Data Analytics

Assignment 3

Zoe Yow Cui Yi | 33214476

Table of Contents

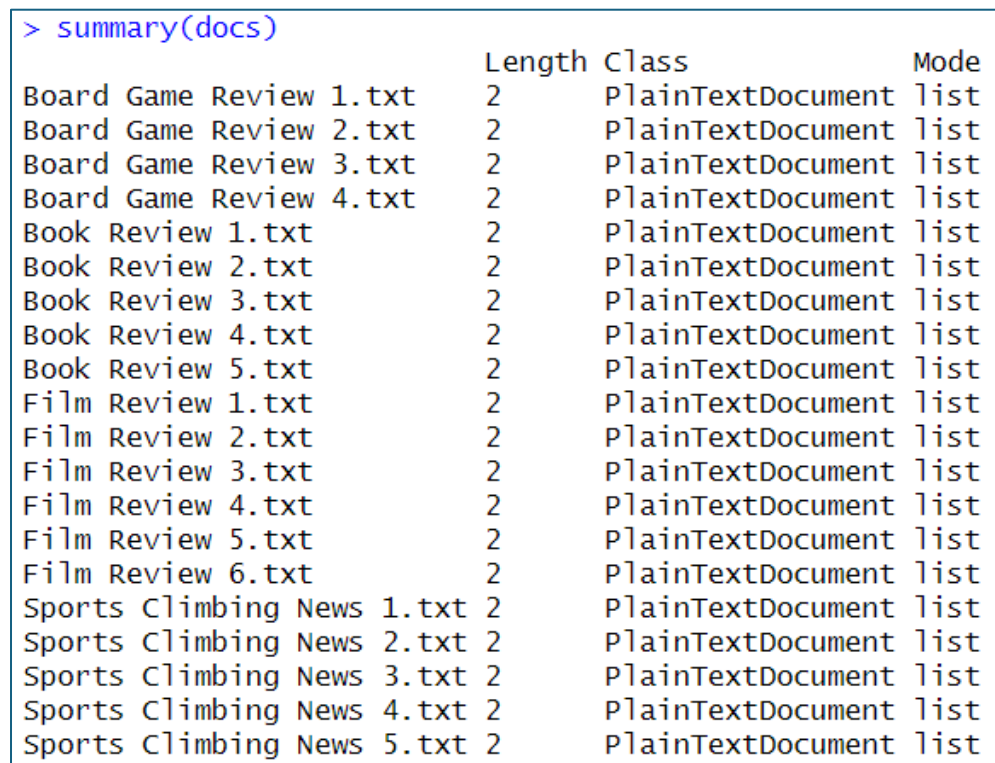
Task 1	2
Task 2	2
Task 3	2
Task 4	3
Task 5	5
Task 6	8
Task 7	13
Task 8	18
Task 9	20
References	21
Appendix	22
Appendix A	22
Appendix B	22
Appendix C	23
Appendix D	23
Appendix E	24

Task 1

For this task, I have chosen documents from 4 different genres, each genre having 4-6 documents. These genres include film reviews, book reviews, sport climbing news, and board game reviews. The documents collected were copied texts, and the URLs for all documents are included in the References.

Task 2

The documents were put into plain text form for processing. To do so, the copied texts were saved as .txt files. Thus, the corpus was created as a folder of .txt files. The figure below shows a summary of the corpus created. As you can see, the length, class, and mode for all documents are 2, "PlainTextDocument", and list, respectively.



```
> summary(docs)
```

	Length	Class	Mode
Board Game Review 1.txt	2	PlainTextDocument	list
Board Game Review 2.txt	2	PlainTextDocument	list
Board Game Review 3.txt	2	PlainTextDocument	list
Board Game Review 4.txt	2	PlainTextDocument	list
Book Review 1.txt	2	PlainTextDocument	list
Book Review 2.txt	2	PlainTextDocument	list
Book Review 3.txt	2	PlainTextDocument	list
Book Review 4.txt	2	PlainTextDocument	list
Book Review 5.txt	2	PlainTextDocument	list
Film Review 1.txt	2	PlainTextDocument	list
Film Review 2.txt	2	PlainTextDocument	list
Film Review 3.txt	2	PlainTextDocument	list
Film Review 4.txt	2	PlainTextDocument	list
Film Review 5.txt	2	PlainTextDocument	list
Film Review 6.txt	2	PlainTextDocument	list
Sports Climbing News 1.txt	2	PlainTextDocument	list
Sports Climbing News 2.txt	2	PlainTextDocument	list
Sports Climbing News 3.txt	2	PlainTextDocument	list
Sports Climbing News 4.txt	2	PlainTextDocument	list
Sports Climbing News 5.txt	2	PlainTextDocument	list

Figure 1: Corpus Summary

Task 3

As part of the text preprocessing for building the Document-Term Matrix (DTM), I applied several transformations to clean and standardize the text following Week 10 Applied. First, I removed numbers and punctuation and converted all text to lowercase to ensure consistency and avoid treating the same word in different cases as separate terms.

Next, I built a custom function using “gsub()” to manually remove unwanted characters and formatting artefacts like different types of quotation marks (‘ “ ” ’), hyphens (-), and possessive endings (’s). These characters appeared frequently in the original text and could interfere with tokenization or cause irregularities in the term list.

I also removed common English stop words, which helped eliminate non-informative words like “the” and “and”. After that, I applied “stripWhitespace” to remove extra spaces introduced during the cleaning process. Finally, I performed stemming to reduce words to their root form (e.g., “running” becomes “run”), which helps group similar terms together.

To control the number of tokens in the final DTM, I adjusted the sparsity threshold through trial and error. I found that a threshold of 0.45 resulted in 20 tokens, which is slightly below the requirement. However, increasing the threshold by even 0.001 led to a significant jump to 36 tokens, which exceeded the target range. Because 0.45 gave a result that was closer to the expected number, I decided to stick with that value. This approach helped retain only the most meaningful and frequently occurring terms in the dataset, ensuring a focused and manageable feature set for analysis.

The DTM created for this task can be found in Appendix A.

Task 4

Figures 2, 3, and 4 below show the dendrogram, the confusion matrix, and the clustering accuracy based on the cosine distance between documents.

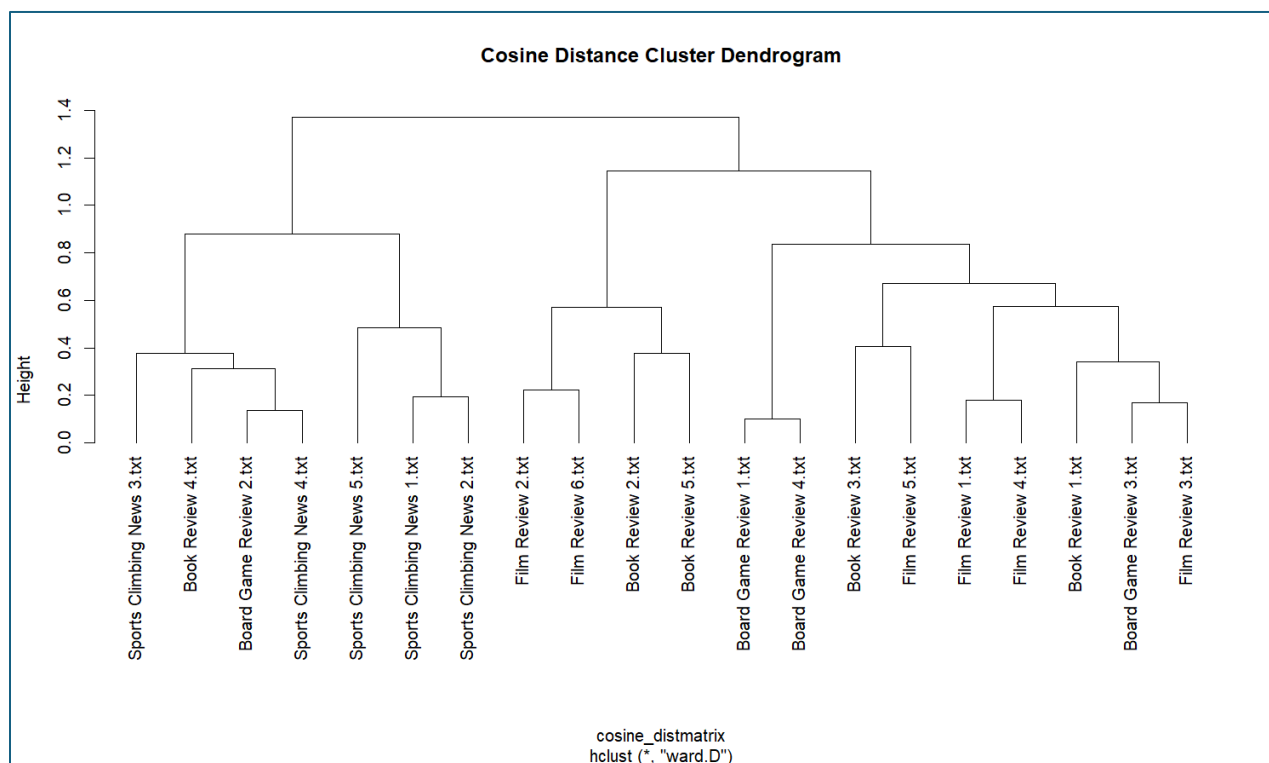


Figure 2: Cosine Distance Cluster Dendrogram

```
> cosine_cf
```

	cosine_groups			
topics	2	1	3	4
BG review	1	1	2	0
book review	1	4	0	0
film review	0	4	2	0
SC news	2	0	0	3

Figure 3: Cosine Distance Cluster Dendrogram Confusion Matrix

```
> cosine_accuracy  
[1] 0.5
```

Figure 4: Cosine Distance Cluster Dendrogram Accuracy

The cosine distance cluster dendrogram provides a generally coherent and interpretable representation of the textual similarities among the documents. Based on the hierarchical structure, the clustering appears to be of good quality, successfully grouping documents with similar thematic content and linguistic patterns.

One example of effective clustering can be observed in the leftmost branch, where all Sports Climbing News articles, namely Sports Climbing News 3, Sports Climbing News 4, Sports Climbing News 5, and Sports Climbing News 1, are grouped closely together. This close proximity at a low height suggests a high degree of textual similarity among these documents, likely due to shared domain-specific vocabulary such as “climber”, “competition”, “boulder”, or “world cup”. Interestingly, this cluster also includes Book Review 4 and Board Game Review 2, indicating that while these documents originate from different categories, they may share common evaluative or descriptive language that aligns with the tone and style of the sports news articles. This could be due to the use of similar expressions such as “performance” or “challenge”.

Another example of meaningful grouping is found near the center of the dendrogram, where Board Game Review 5, Book Review 3, and Board Game Review 1 are clustered together. These documents likely share a common structure focused on reviewing recreational content, with overlapping terminology related to features such as “gameplay”, “mechanisms”, or “fun”. The inclusion of Film Review 5 in this group suggests that although the subject matter differs, the document may follow a similar critique format, utilizing language associated with narrative analysis or audience impact.

On the right side of the dendrogram, a tight cluster consisting of Film Review 1, Film Review 3, and Film Review 4 documents further supports the quality of the clustering. These documents are likely to be thematically consistent, using terminology associated with film analysis such as “cinematography”, “plot”, “acting”, or “direction”. The presence of Book Review 1 and Board Game Review 3 within this cluster may reflect the use of similar critical language across reviews of different media types, such as discussions of storyline.

To further evaluate the quality of the clustering, a confusion matrix was constructed using the known document categories. From this, an accuracy score of 0.5 was obtained. This value indicates that approximately half of the documents were correctly grouped in line with their original categories. While this reflects a moderate level of alignment, it also suggests that there is considerable overlap in language and writing style across some document types, which may have affected the clustering outcome. In particular, the use of cosine distance emphasizes similarity in the direction of term usage, meaning documents with similar descriptive or analytical language can be grouped together even if their subjects differ.

In conclusion, the dendrogram effectively captures meaningful relationships among documents based on their linguistic structure and thematic elements. While not perfectly aligned with the original labels, the clustering shows sufficient coherence and interpretability. The moderate accuracy score highlights both the potential and limitations of unsupervised clustering using cosine distance, especially in datasets where cross-category linguistic overlap is present.

Task 5

The sentiment analysis was conducted following Week 11 Lecture, using the “SentimentAnalysis” package in R. The “analyzeSentiment()” function was applied to the document corpus to generate sentiment scores. The output was a variety of measures from all dictionaries, including “DictionaryGI”, “DictionaryHE”, “DictionaryLM”, and “DictionaryQDAP”. The word counts were also extracted. This output was then merged with a manually constructed genre label extracted from the document names. Specifically, I generated a data frame from document identifiers using the “summary()” function and removed duplicate entries to ensure each sentiment score corresponded uniquely to a single document. The genre labels were derived by extracting the first three characters of each document identifier, for example “Fil” for film reviews or “Spo” for sports climbing news, which served as categorical groupings for the sentiment analysis. This output can be found in Appendix B.

Although all dictionaries were used, only specific measures from some of the dictionaries were included in this analysis. This includes “SentimentQDAP”, “PositivityQDAP”, “RatioUncertaintyLM”, and “WordCount”. “SentimentQDAP” and “PositivityQDAP” measure the general sentiment polarity score, and positivity score derived from the QDAP (Quantitative Discourse Analysis Package) dictionary respectively. Moreover, “RatioUncertaintyLM” measures the lexical uncertainty based on the LM (Loughran-McDonald) financial sentiment dictionary. Lastly, “WordCount” measures the total number of words in each document.

To explore the differences in sentiment across genres, I visualized the sentiment metrics using boxplots. Each plot displayed the distribution of a sentiment variable grouped by genre, enabling comparative analysis across different types of texts. These visualizations were saved as a single PDF file and are shown below in Figure 5.

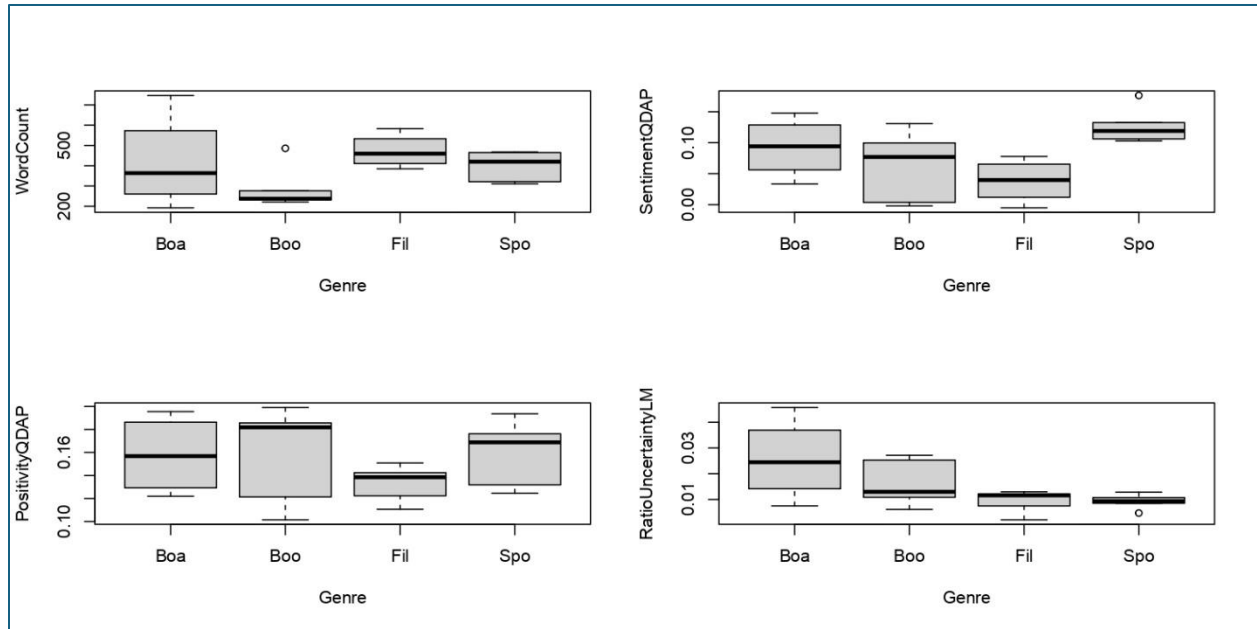


Figure 5: Box Plots of Several Measures of the Sentiment Analysis for Genres

The boxplots reveal notable differences in both the average sentiment and the variability of sentiment-related metrics across genres. For instance, sports climbing news documents exhibit higher average “SentimentQDAP” scores, while film reviews show consistently lower “PositivityQDAP”. Additionally, board game reviews show greater variability in both “WordCount” and “RatioUncertaintyLM”, suggesting a more diverse writing style or range of expression.

To determine whether the observed differences in sentiment measures between genres were statistically significant, pairwise independent t-tests were conducted for each combination of genres across three sentiment-related variables: “PositivityQDAP”, “SentimentQDAP”, and “RatioUncertaintyLM”.

Figure 6 below shows the results of the pairwise independent t-test for the “PositivityQDAP” variable with a null hypothesis that a genre has less or similar positivity than another genre.

```
> p_vals_df[order(p_vals_df$p_values),]
      genre p_values
2  boa and fil 0.1316074
4  boo and fil 0.1467585
1  boa and boo 0.5017193
5  boo and spo 0.5182315
3  boa and spo 0.5217161
6  fil and spo 0.9310203
```

Figure 6: Pairwise Independent T-test Results for “PositivityQDAP”

Based on the results, the observed differences in “PositivityQDAP” between genres are not statistically significant. This is because all resulting p-values exceed the 0.05 threshold, providing weak evidence against the null hypothesis. Therefore, the analysis suggests that the level of positivity is relatively similar across genres, with no genre exhibiting significantly higher or lower positivity than another.

Moving on, Figure 7 below shows the results of the pairwise independent t-test for the “SentimentQDAP” variable with the null hypothesis that a genre can have greater or similar sentiment than another genre.

```
> p_vals_df[order(p_vals_df$p_values),]
      genre    p_values
6 fil and spo 0.0005004143
5 boo and spo 0.0343153324
3 boa and spo 0.1320482325
4 boo and fil 0.7733583871
1 boa and boo 0.7902740766
2 boa and fil 0.9459630287
```

Figure 7: Pairwise Independent T-test Results for “SentimentQDAP”

Based on the results, it can be concluded that most genres do not exhibit lower sentiment compared to others. This is supported by p-values greater than 0.05, which provide weak evidence against the null hypothesis of equal sentiment. However, Sports Climbing News demonstrates significantly higher sentiment than both Book Reviews and Film Reviews, with p-values of 0.03432 and 0.0005004, respectively. These values provide strong and very strong evidence against the null hypothesis, particularly the latter, which indicates a highly significant difference in sentiment levels.

Lastly, Figure 8 below shows the results of the pairwise independent t-test for the “RatioUncertaintyLM” variable with the null hypothesis that a genre can have less or similar linguistic uncertainty than another genre.

```
> p_vals_df[order(p_vals_df$p_values),]
      genre    p_values
3 boa and spo 0.06602427
2 boa and fil 0.06925444
5 boo and spo 0.07920582
4 boo and fil 0.09167206
1 boa and boo 0.18171850
6 fil and spo 0.42632651
```

Figure 8: Pairwise Independent T-test Results for “RatioUncertaintyLM”

Based on the results, none of the genres exhibit significantly higher levels of linguistic uncertainty compared to others. Although Board Game Reviews display a higher average level of linguistic

uncertainty than Sports Climbing News and Film Reviews, the associated p-values exceed the 0.05 significance threshold. This provides only weak evidence against the null hypothesis. Therefore, it can be concluded that all genres demonstrate relatively similar levels of linguistic uncertainty.

Task 6

For this task, I created the basic single mode network following the method shown in Week 11. Figure 9 below shows the network created between the documents based on the number of shared terms.

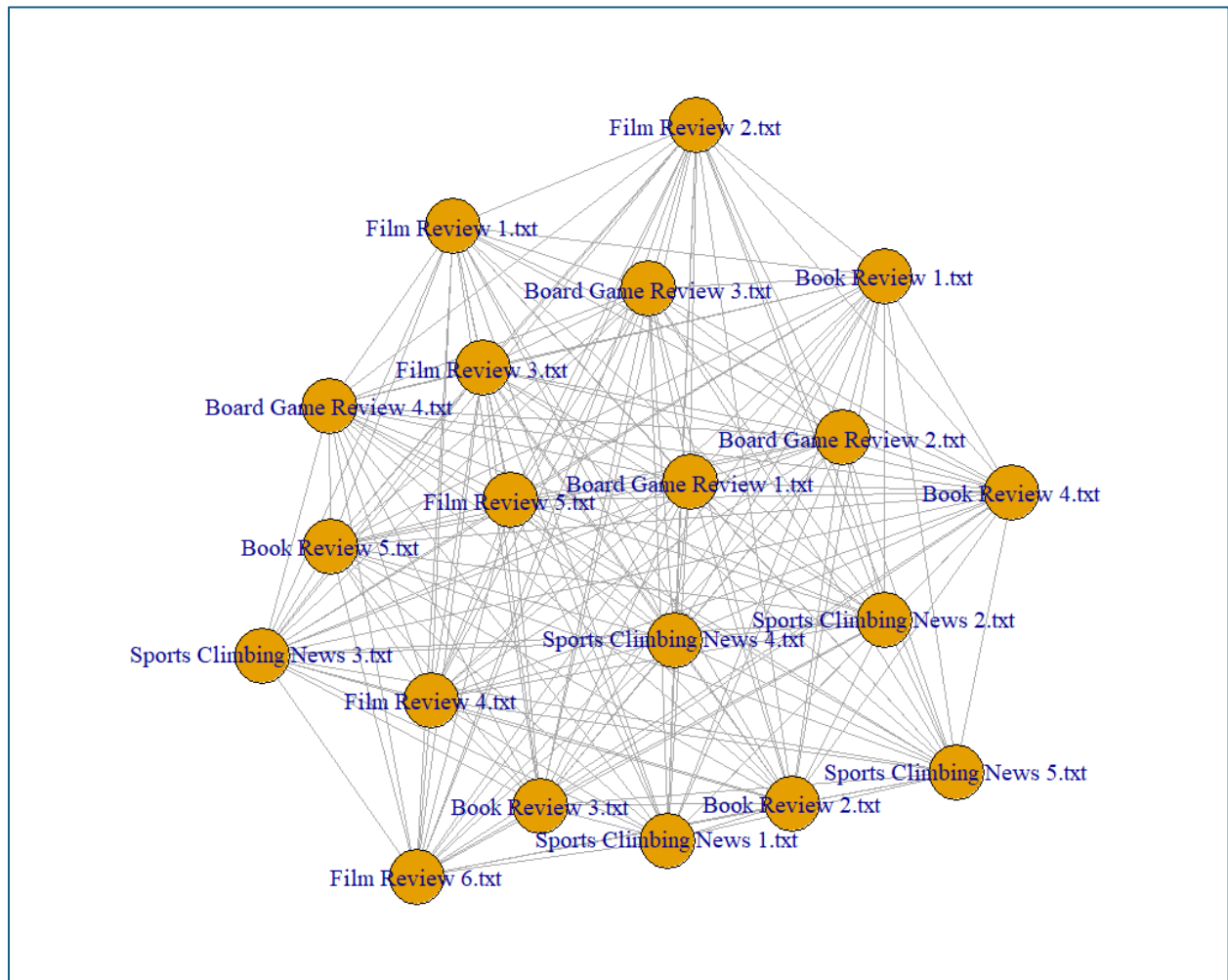


Figure 9: Basic Single-Mode Network for Documents

This figure is not highly informative, clear relationships between documents and groups cannot be identified just by looking at this basic network plot. It does, however, function as a preliminary visualisation that enables me to spot possibilities for a later, more thorough, and instructive network plot.

The network centralities were measured for each document, and it can be found in Appendix C. Figure 10, 11, 12, and 13 show the top 5 documents sorted by each centrality measure values.

```
> head(stats[order(-stats$betweenness),])
```

	degree	betweenness	closeness	eigenvector
Film Review 2.txt	19	1.5	0.007246377	0.5757476
Board Game Review 1.txt	19	0.0	0.004048583	1.0000000
Board Game Review 2.txt	19	0.0	0.004901961	0.8379064
Board Game Review 3.txt	19	0.0	0.004784689	0.8571262
Board Game Review 4.txt	19	0.0	0.004255319	0.9546295
Book Review 1.txt	19	0.0	0.005649718	0.7295729

Figure 10: Top 5 Documents Sorted by Betweenness Value

```
> head(stats[order(-stats$closeness),])
```

	degree	betweenness	closeness	eigenvector
Film Review 2.txt	19	1.5	0.007246377	0.5757476
Film Review 5.txt	19	0.0	0.006329114	0.6563197
Sports Climbing News 5.txt	19	0.0	0.006097561	0.6807139
Book Review 4.txt	19	0.0	0.005681818	0.7285363
Book Review 1.txt	19	0.0	0.005649718	0.7295729
Book Review 5.txt	19	0.0	0.005524862	0.7477934

Figure 11: Top 5 Documents Sorted by Closeness Value

```
> head(stats[order(-stats$eigenvector),])
```

	degree	betweenness	closeness	eigenvector
Board Game Review 1.txt	19	0	0.004048583	1.0000000
Film Review 3.txt	19	0	0.004255319	0.9547403
Board Game Review 4.txt	19	0	0.004255319	0.9546295
Sports Climbing News 4.txt	19	0	0.004329004	0.9402908
Film Review 4.txt	19	0	0.004629630	0.8829968
Sports Climbing News 1.txt	19	0	0.004716981	0.8685629

Figure 12: Top 5 Documents Sorted by Eigenvector Value

```
> head(stats[order(-stats$degree),])
```

	degree	betweenness	closeness	eigenvector
Board Game Review 1.txt	19	0	0.004048583	1.0000000
Board Game Review 2.txt	19	0	0.004901961	0.8379064
Board Game Review 3.txt	19	0	0.004784689	0.8571262
Board Game Review 4.txt	19	0	0.004255319	0.9546295
Book Review 1.txt	19	0	0.005649718	0.7295729
Book Review 2.txt	19	0	0.004784689	0.8580900

Figure 13: Top 5 Documents Sorted by Degree Value

The degree centrality for all documents is identical, indicating that each document has the same number of direct connections. As such, this measure does not contribute meaningfully to

identifying which documents are most central or influential within the network and can be excluded from further analysis.

An inverse relationship appears to exist between closeness centrality and eigenvector centrality. As closeness decreases, eigenvector values tend to increase. This suggests that documents more tightly connected to highly influential nodes (reflected by high eigenvector scores) may not necessarily be the most centrally located in terms of distance to other documents.

A positive relationship is observed between closeness and betweenness centrality. Documents with higher closeness values tend to also exhibit higher betweenness, suggesting that central positioning within the network corresponds with a greater role in bridging different parts of the network. This is particularly evident in Film Review 2, which has the highest betweenness and closeness values but the lowest eigenvector centrality. This implies that while it serves as a key connector or pathway in the network, it is not strongly connected to other highly influential documents.

In contrast, Board Game Review 1 exhibits the highest eigenvector centrality, indicating strong association with influential nodes, despite having one of the lowest closeness scores. Similarly, Board Game Review 3 and Film Review 3 both have high eigenvector values and low closeness values, suggesting they are connected to important nodes but are not centrally located themselves. Sports Climbing News 1 presents a balance, with a high eigenvector value and an average closeness score, implying both influence and moderate centrality.

In conclusion, the most important documents in the network vary depending on the aspect of centrality considered. Film Review 2 is the most central in terms of facilitating connections and proximity (betweenness and closeness), whereas Board Game Review 1 is the most influential in terms of its connection to other well-connected documents (eigenvector). These documents represent different types of importance: Film Review 2 for structural connectivity, and Board Game Review 1 for network influence.

To improve the graph over the basic single-mode network shown in Figure 9, the edge betweenness and the fast greedy methods were applied to the network. They are shown in Figure 14 and 15.

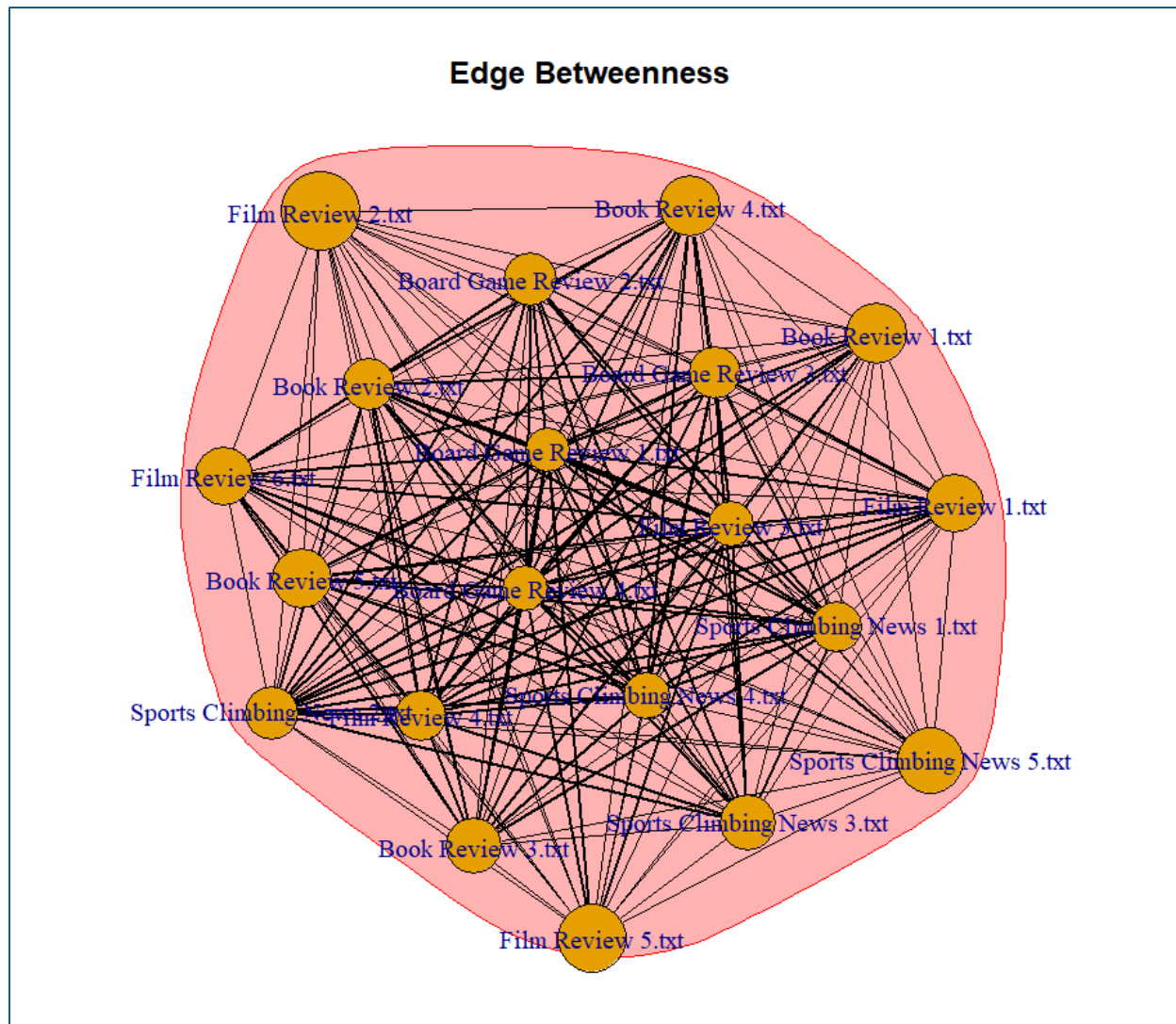


Figure 14: Documents Single-mode Network with Edge Betweenness Algorithm

Based on the figure above, the importance of each vertex is indicated by its size. The larger vertices signify more central or influential documents according to their closeness centrality. Stronger connections between vertices are indicated by darker lines, which emphasize texts with more words in common. It can be said that only a single group was discovered using the “cluster_edge_betweenness” group discovery technique. This suggests that there are not any further unique groupings and majority of documents are strongly related to one another, indicating that the document network's general structure is quite logical. The phrases used in the various genres are not sufficiently distinctive from one another to create discrete groups within the network.

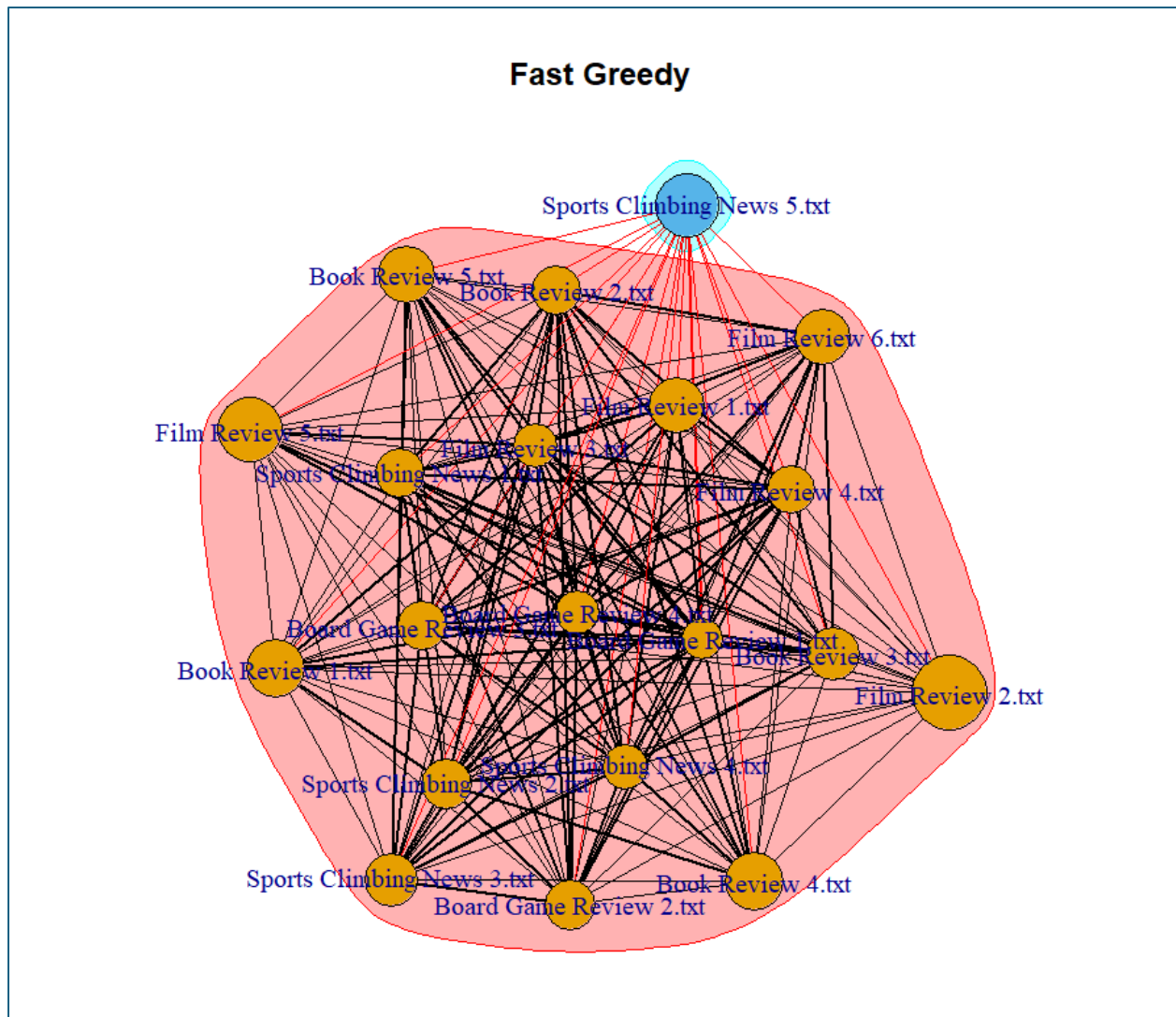


Figure 15: Documents Single-mode Network with Fast Greedy Algorithm

Based on the figure above, it follows similar logic as Figure 14, where the importance of each vertex is indicated by its size, and stronger connections between vertices are indicated by darker lines. However, vertices with a different colour, like Sports Climbing News 5's blue, indicate that they are part of a different group.

Since Sports Climbing News 5 is in another group than the other documents, it has a distinct sequence of word connections than the other documents in the network. The "cluster_fast_greedy" group discovery method produced this unique group identity. However, every document except Sports Climbing News 5 has comparable structures, indicating that they are part of the same group and use more common words. This suggests that these documents have similar structural patterns within the network and overlap word usage, even if they belong to various genres (Book Review, Board Game Review, Film Review, and Sports Climbing News).

Task 7

For this task, I also created the basic single mode network following the method shown in Week 11. Figure 16 below shows the network created between the tokens.

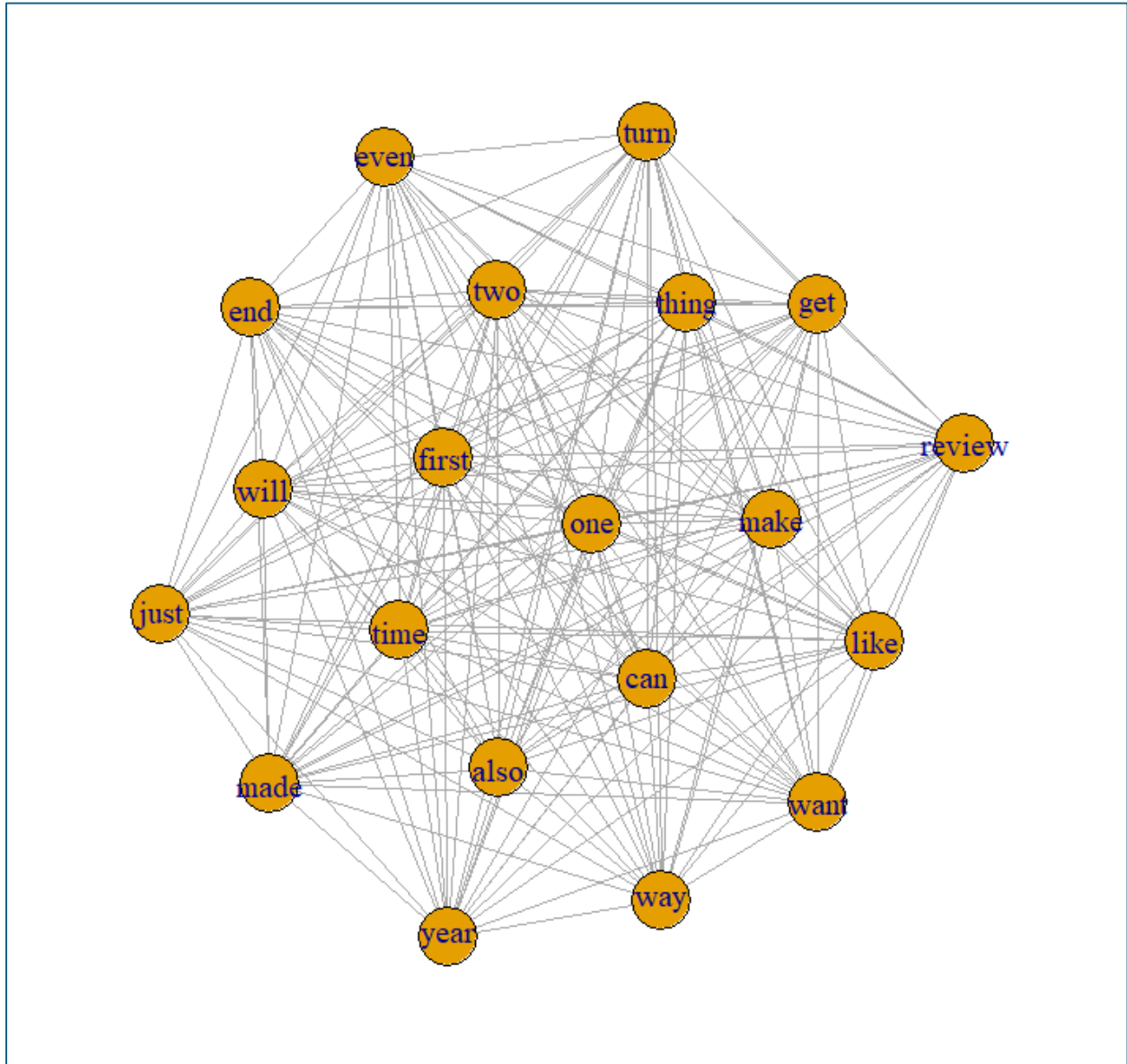


Figure 16: Basic Single-mode Network for Tokens

Similarly to Task 6, this figure is not highly informative, clear relationships between documents and groups cannot be identified just by looking at this basic network plot. It does, however, function as a preliminary visualisation that enables me to spot possibilities for a later, more thorough, and instructive network plot.

The network centralities were measured for each token, and it can be found in Appendix D. Figure 17, 18, 19, and 20 show the top 5 documents sorted by each centrality measure values.

```
> head(stats_token[order(-stats_token$betweenness),])
```

	degree	betweenness	closeness	eigenvector
also	19	0	0.004587156	0.9173062
can	19	0	0.004219409	0.9871138
end	19	0	0.005586592	0.7583253
first	19	0	0.004273504	0.9738972
get	19	0	0.004975124	0.8463451
just	19	0	0.005494505	0.7696246

Figure 17: Top 5 Tokens Sorted by Betweenness Value

```
> head(stats_token[order(-stats_token$closeness),])
```

	degree	betweenness	closeness	eigenvector
year	19	0	0.006134969	0.6911888
review	19	0	0.006097561	0.6969206
turn	19	0	0.006060606	0.6980473
even	19	0	0.006060606	0.6971119
end	19	0	0.005586592	0.7583253
thing	19	0	0.005555556	0.7592578

Figure 18: Top 5 Tokens Sorted by Closeness Value

```
> head(stats_token[order(-stats_token$eigenvector),])
```

	degree	betweenness	closeness	eigenvector
one	19	0	0.004166667	1.0000000
can	19	0	0.004219409	0.9871138
first	19	0	0.004273504	0.9738972
time	19	0	0.004329004	0.9647623
also	19	0	0.004587156	0.9173062
make	19	0	0.004629630	0.9088714

Figure 19: Top 5 Tokens Sorted by Eigenvector Value

```
> head(stats_token[order(-stats_token$degree),])
```

	degree	betweenness	closeness	eigenvector
also	19	0	0.004587156	0.9173062
can	19	0	0.004219409	0.9871138
end	19	0	0.005586592	0.7583253
first	19	0	0.004273504	0.9738972
get	19	0	0.004975124	0.8463451
just	19	0	0.005494505	0.7696246

Figure 20: Top 5 Tokens Sorted by Degree Value

The degree centrality and betweenness centrality values are identical for all tokens, indicating that each token has an equal number of direct connections and does not serve as a bridge in the network. As a result, these two measures can be excluded from determining the most important tokens in the network.

As observed in Task 6, there appears to be an inverse relationship between eigenvector centrality and closeness centrality. This pattern is clearly illustrated by the tokens “year” and “one.” The token “year” has the highest closeness centrality, suggesting it is centrally positioned and can reach other tokens efficiently, but it has the lowest eigenvector centrality, indicating it is not well-connected to other influential tokens. Conversely, the token “one” has the highest eigenvector centrality and the lowest closeness centrality, implying that although it is less centrally located, it is well-connected to other prominent tokens.

The tokens “turn” and “even” both have high closeness centrality and moderate eigenvector centrality, showing that they are fairly central in the network and moderately connected to influential tokens, although they do not dominate either metric.

Therefore, the most important tokens in the network vary depending on the aspect of importance being considered. The token “year” is most central in terms of accessibility and reach (closeness), while “one” holds greater influence due to its connection to other highly connected tokens (eigenvector). These findings suggest that different tokens contribute to the structure of the text network in distinct ways, either through central positioning or through influential associations.

Similarly to Task 6, to improve the graph over the basic single-mode network shown in Figure 16, the edge betweenness and the fast greedy methods were applied to the network. They are shown in Figure 21 and 22.

Edge Betweenness For Tokens

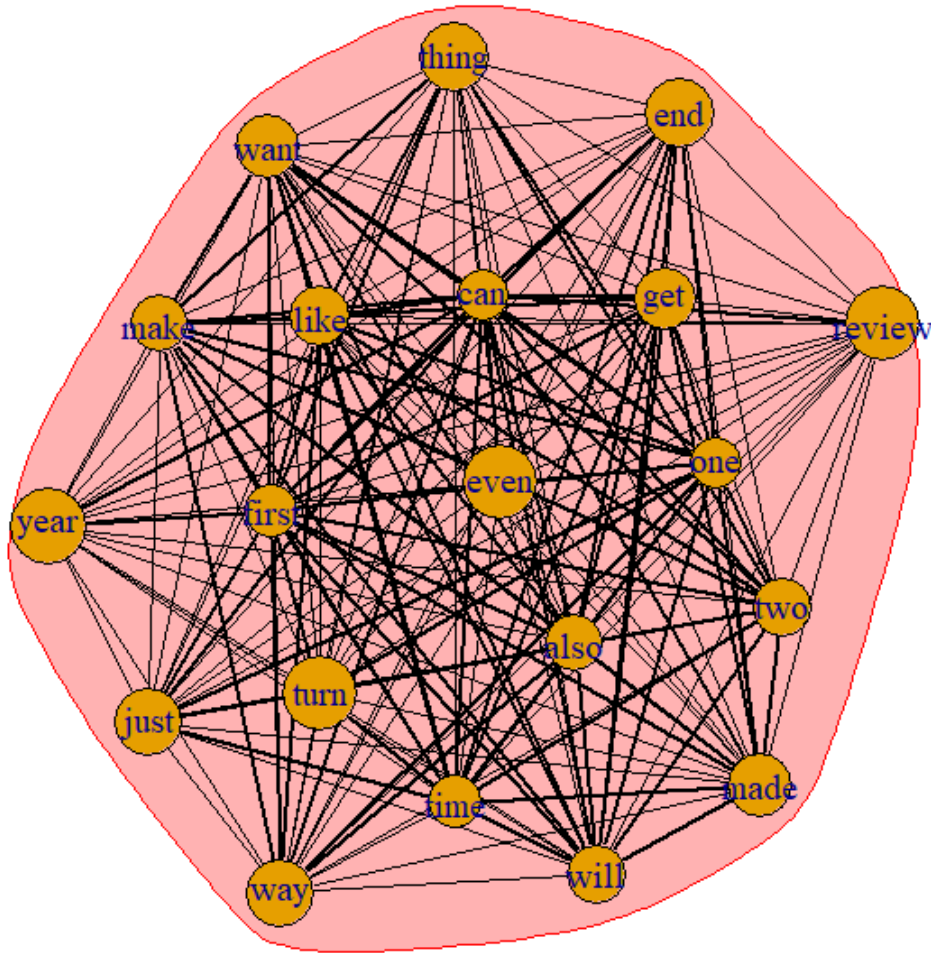


Figure 21: Tokens Single-mode Network with Edge Betweenness Algorithm

Based on the figure above, the importance of each vertex is also indicated by its size. The larger vertices signify more central or influential tokens according to their closeness centrality. Stronger connections between vertices are also indicated by darker lines, which emphasize tokens with more documents in common.

Using the edge betweenness method, which identified one group, this can be explained the same way as the document edge betweenness network (Figure 14). It displays a unified structure with a single cluster. This implies that a single, close-knit group has formed because of the tokens in the corpus sharing similar patterns of overlap. The network's closeness suggests that the tokens are intricately linked to one another and have little distance from one another.

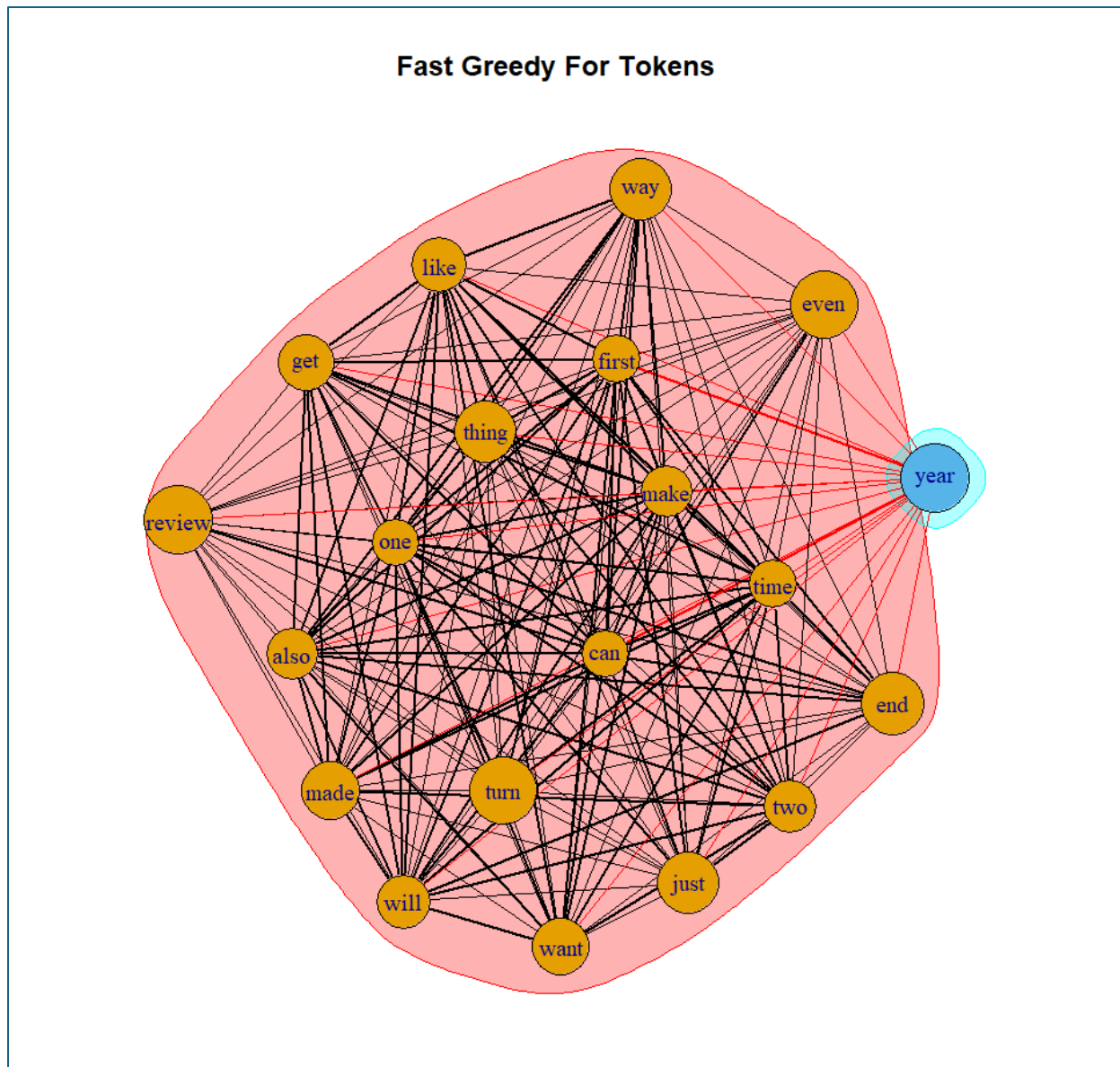


Figure 22: Tokens Single-mode Network with Fast Greedy Algorithm

This improved graph, like Figure 21, illustrates the significance of vertices through their size, where larger vertices represent higher centrality. The colour of the edges indicates the different groups assigned to the vertices based on community detection, while the thickness of the edges represents the strength of the connections. Edges with different colours connecting nodes from separate groups highlight inter-group relationships. Notably, two distinct groups emerge from the graph.

Only one token (i.e., “year”) appears in a different group from all the other tokens. This may suggest that “year” plays a unique or specialized role in the text network, possibly representing a distinct semantic or contextual usage that does not align closely with the other tokens. Its separation could also indicate that it functions in a different thematic context or appears in different

types of documents, reflecting a pattern that distinguishes it from the rest of the corpus. This structural isolation supports previous findings from centrality analysis, where “year” exhibited high closeness centrality but low eigenvector centrality, reinforcing its role as a central yet independently functioning token in the network.

Task 8

For this task, I also created the bipartite network following the method shown in Week 11. Figure 23 below shows the network created of the corpus, with document ID as one type of node and tokens as the other type of node.

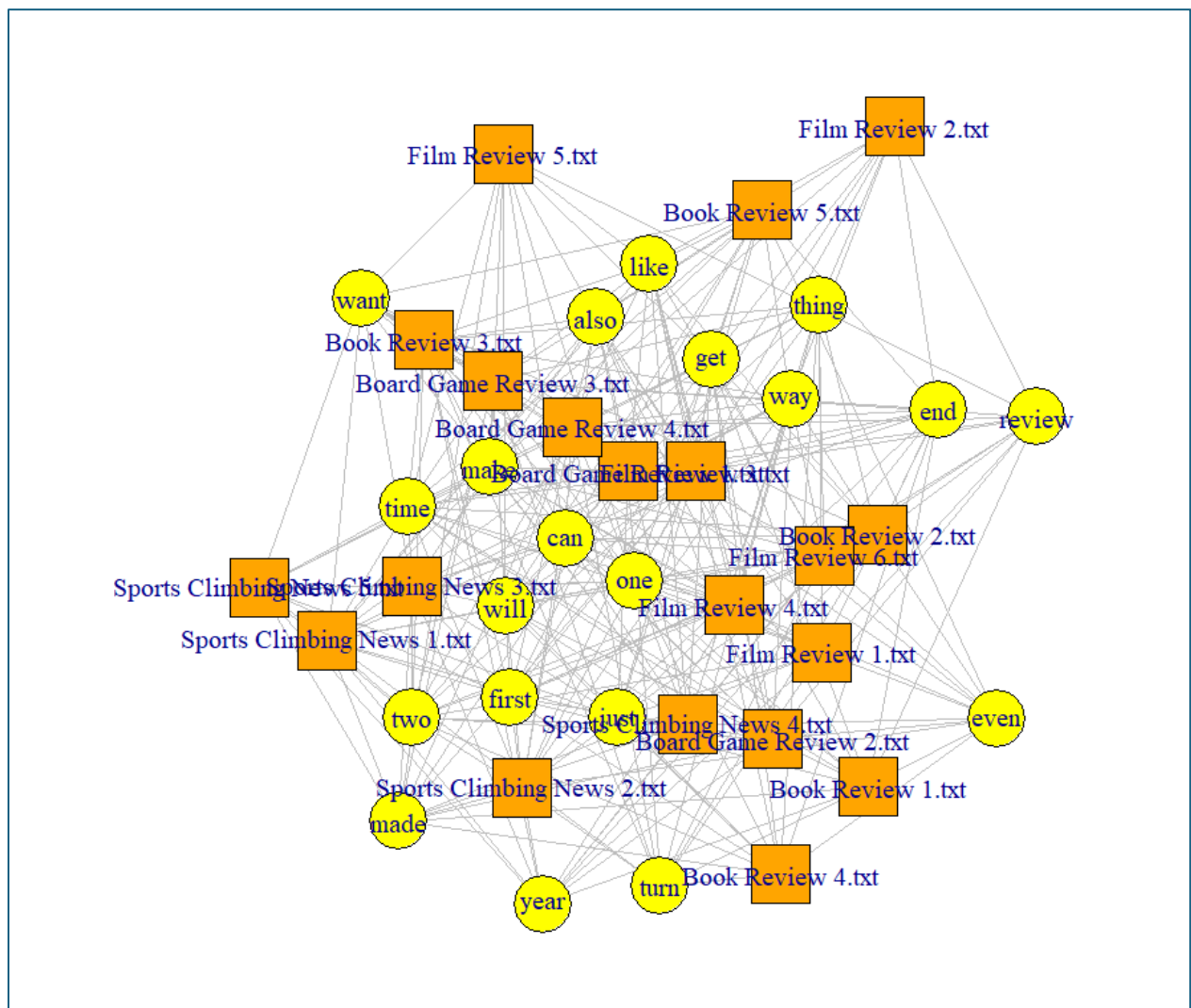


Figure 23: Bipartite Network of Corpus

This figure is not highly informative, clear relationships between documents and groups cannot be identified just by looking at this network. A more thorough and instructive network plot is shown below as Figure 24.

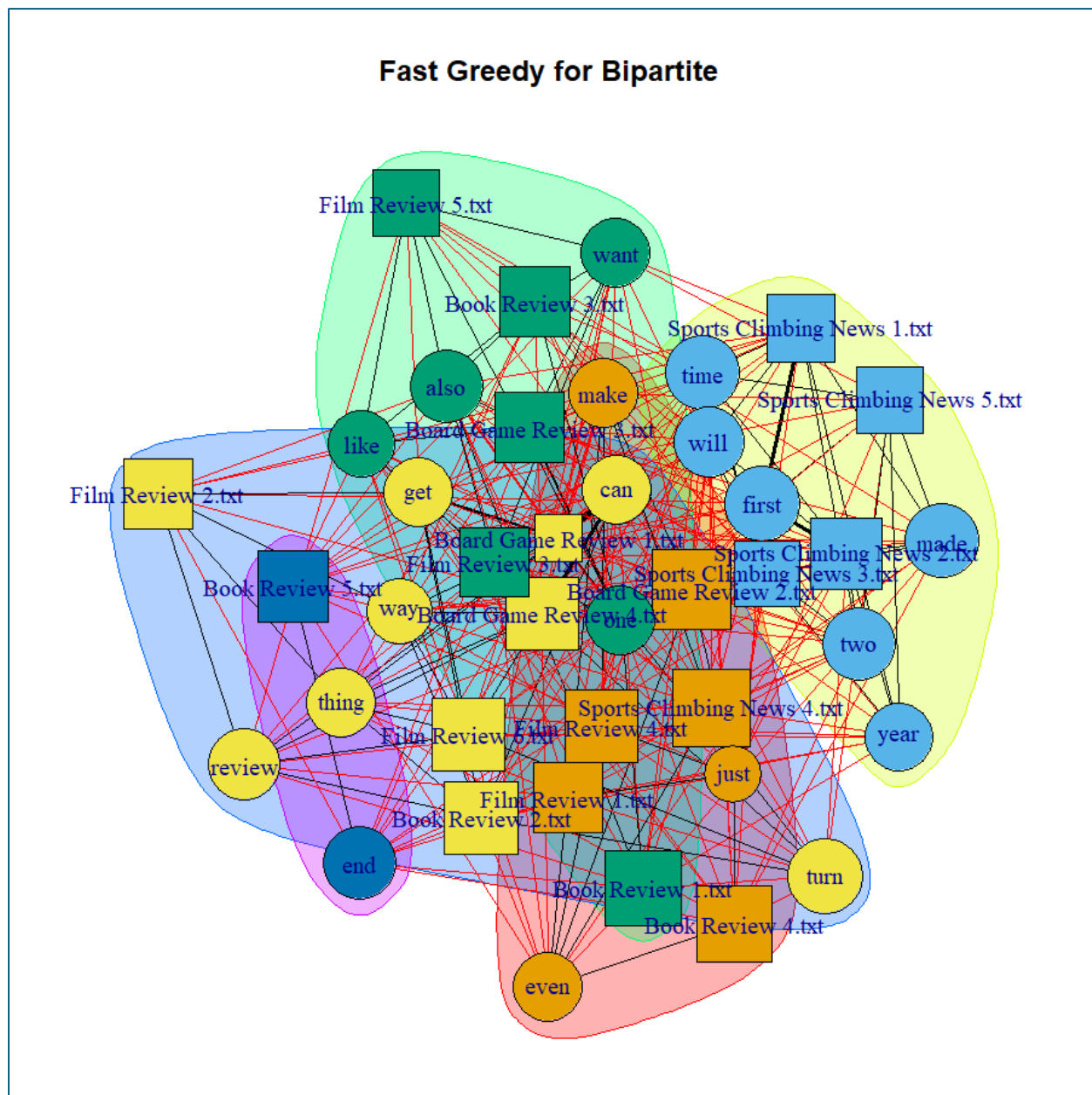


Figure 24: Bipartite Network with Fast Greedy Algorithm

This graph visualizes the relationship between documents and words using a bipartite network structure, analysed with the Fast Greedy community detection algorithm. In this network, square nodes represent individual documents, while circular nodes represent tokens. Edges between nodes indicate the presence of a word in a given document. Moreover, the importance of each vertex is also indicated by its size. The larger vertices signify more central or influential documents/tokens according to their closeness centrality. Stronger connections between vertices are also indicated by darker lines.

The use of colour and shaded regions in the graph highlights distinct communities or clusters that have been automatically detected based on the density of connections. These communities suggest groups of documents that share a similar vocabulary, and likewise, groups of words that frequently co-occur in the same types of documents.

Several clear groupings can be identified:

- Documents related to film reviews (e.g., Film Review 2.txt, Film Review 6.txt) are clustered together with words like review, thing, and get.
- Sports climbing news articles (e.g., Sports Climbing News 1.txt, Sports Climbing News 5.txt) form another distinct group, associated with words like first, time, year, and made.
- Green, red, and purple clusters contains book reviews and board game reviews, clustered around words such as also, want, even, end, and like.

This structure demonstrates that the Fast Greedy algorithm is effective in identifying meaningful groupings within the data, allowing for an intuitive understanding of the thematic similarities among documents based on shared vocabulary. Overall, the graph provides a useful visual summary of the semantic relationships within the corpus.

Task 9

To summarise, the important documents include Film Review 2 and Board Game Review 1, while the important tokens include “one” and “year”. Moreover, based on the bipartite graph, the most important group is the yellow cluster. This cluster demonstrates a high degree of connectivity, with numerous edges linking key documents such as Sports Climbing News 1.txt, News 3.txt, and News 5.txt to a variety of word tokens. The words in this cluster, including “year,” “first,” “time,” and “will,” are relatively general yet highly connected, suggesting they play a central role in shaping the semantic structure of the entire graph. The documents within the yellow cluster are not only densely interconnected but also positioned centrally, indicating their influence across different parts of the network. These features collectively suggest that the yellow cluster is the most significant in terms of both structural importance and thematic relevance within the network.

Clustering is an effective method for identifying broad patterns in data by grouping similar items based on shared characteristics. In the context of the bipartite graph, clustering helps to reveal distinct communities of documents and words that are more closely related to each other. This makes it easier to detect thematic groups, such as sports news or film reviews, and to simplify complex datasets into more interpretable categories.

Network analysis, in contrast, provides a more detailed understanding of the relationships between individual elements within the data. It can identify central or influential nodes, such as frequently used words or key documents, and uncover how these elements connect different groups. Tools like degree centrality and edge density offer insights into which words or documents are most important, either within their clusters or across the entire network.

Overall, clustering and network analysis serve complementary roles. Clustering is useful for gaining a high-level view of the data and discovering overarching themes, while network analysis

is valuable for exploring the finer structure and pinpointing influential nodes. When used together, they provide a well-rounded and comprehensive analysis of relationships and importance within the dataset.

To enhance text processing and improve the ability to differentiate between documents, incorporating n-grams, such as bigrams or trigrams, allows the model to capture commonly occurring word combinations and contextual phrases. This is especially useful for identifying domain-specific language patterns. Additionally, the use of Named Entity Recognition (NER) can enhance document distinction by extracting and categorizing proper nouns such as names, places, or events. These enhancements collectively lead to a more robust and semantically rich representation of the text, ultimately improving the accuracy of clustering and network analysis in identifying important groups and relationships within the dataset.

References

Board Game Reviews:

- <http://www.gbreviews.com/board-game-review-codenames/>
- https://www.board-game.co.uk/monopoly-review/?srsId=AfmBOoqGbKgZ7ba_7aHcAjdAMvFtT2bJQnc_WF1eXk9Vs2xaERwCXwok
- <https://geektogeekmedia.com/geekery/clue-rivals-edition-the-board-game-review/>
- <https://www.board-game.co.uk/articulate-review/?srsId=AfmBOophQ6WdMnFhbHU5WwtbkSGvXuy7ZjonIdjHvfYPSKmDwQ1W8FLX>

Book Reviews:

- <https://tosinadeoti.medium.com/animal-farm-by-george-orwell-7de08e200bb1>
- <https://author-ashok.medium.com/book-review-the-boy-in-the-striped-pyjamas-9e4c69d0de5b>
- <https://katherinehamadeh.com/2024/01/09/the-summer-i-turned-pretty-book-review/>
- <https://www.theguardian.com/childrens-books-site/2015/nov/05/to-all-the-boys-ive-loved-before-jenny-han-review>
- <https://www.theguardian.com/childrens-books-site/2014/apr/03/review-fault-in-our-stars-john-green>

Film Reviews:

- <https://booksforkeeps.co.uk/article/the-hunger-games-film-review/>
- <https://www.theguardian.com/film/2019/apr/23/avengers-endgame-review-unconquerable-brilliance-takes-marvel-to-new-heights>
- <https://theoxfordblue.co.uk/glass-onion-review/>
- <https://www.theguardian.com/film/2019/jun/14/murder-mystery-review-netflix-adam-sandler-jennifer-aniston>

- <https://www.cinemaescapist.com/2023/04/review-hunger-thai-netflix-movie/>
- <https://www.theguardian.com/film/2017/dec/24/jumanji-welcome-to-the-jungle-mark-kermode-film-of-the-week-review>

Sports Climbing News:

- <https://www.ifsc-climbing.org/news/anraku-makes-it-three-in-a-row-in-salt-lake-city>
- <https://www.ifsc-climbing.org/news/garnbret-survives-injury-scare-for-second-olympic-gold>
- <https://www.ifsc-climbing.org/news/watson-smashes-world-record-on-way-to-bali-gold>
- <https://www.ifsc-climbing.org/news/anraku-triumphs-and-women-share-wujiang-gold>
- <https://www.ifsc-climbing.org/events/ifsc-world-cup-kejiao-2025/news/mackenzie-motivated-and-bertone-back-for-first-final-line-up-of-2025>

Appendix

Appendix A

Task 3 Document-Term Matrix

	also	can	end	frst	get	just	like	made	make	one	review	thing	time	turn	two	want	way	will	even	year	
Board Game Review 1.txt	7	20	2	2	2	7	3	4	2	8	8	2	5	4	2	3	2	4	3	0	0
Board Game Review 2.txt	2	2	0	1	2	4	1	1	1	1	0	1	0	1	0	1	0	1	1	1	1
Board Game Review 3.txt	3	2	1	1	0	0	5	1	2	2	9	1	2	2	1	2	2	0	3	0	0
Board Game Review 4.txt	2	12	1	5	2	0	3	1	2	2	2	1	3	1	2	1	1	2	4	0	0
Book Review 1.txt	1	1	2	2	1	0	0	1	0	5	1	0	1	1	0	0	0	0	1	2	1
Book Review 2.txt	0	1	3	1	2	2	2	0	0	0	3	4	1	2	2	1	2	1	1	1	1
Book Review 3.txt	2	2	0	1	0	0	5	0	3	2	2	0	1	5	1	1	4	1	0	1	1
Book Review 4.txt	0	0	1	2	1	3	1	1	0	2	0	1	1	1	1	1	0	0	1	2	0
Book Review 5.txt	1	1	6	0	1	2	3	0	1	1	1	0	0	1	0	0	2	3	1	1	0
Film Review 1.txt	2	2	2	0	0	3	0	1	5	3	2	1	1	1	1	1	0	0	0	3	1
Film Review 2.txt	1	0	1	1	4	0	1	0	1	1	1	1	2	0	0	0	0	3	0	0	0
Film Review 3.txt	2	2	1	1	4	4	5	1	1	6	2	2	2	3	0	2	3	0	3	2	0
Film Review 4.txt	3	1	0	1	0	3	2	0	3	4	1	1	1	1	1	1	1	2	0	3	3
Film Review 5.txt	6	1	0	0	4	0	3	1	2	1	1	0	1	0	0	2	0	1	0	0	0
Film Review 6.txt	0	1	0	1	3	0	2	0	1	2	1	1	2	0	1	0	1	3	1	1	3
Sports Climbing News 1.txt	1	3	0	10	1	1	1	2	5	1	0	0	0	2	0	1	2	1	3	0	3
Sports Climbing News 2.txt	1	3	1	10	1	1	0	2	0	6	0	0	0	5	1	3	0	1	2	1	2
Sports Climbing News 3.txt	1	2	1	1	0	6	0	3	2	4	0	0	0	7	0	3	2	4	4	0	1
Sports Climbing News 4.txt	1	2	1	2	1	10	1	3	1	1	0	1	1	0	0	2	1	2	2	3	2
Sports Climbing News 5.txt	0	0	0	2	2	2	0	1	3	1	0	1	2	1	4	3	0	0	0	0	2

Appendix B

Task 5 Sentiment Analysis Measures CSV File

	Genre	WordCount	SentimentGI	NegativityGI	PositivityGI	SentimentHE	NegativityHE	PositivityHE	SentimentLM	NegativityLM	PositivityLM	RatioUncertaintyLM	SentimentQDAP	NegativityQDAP	PositivityQDAP
1	Boa	747	0.103078983	0.111111111	0.214190094	0.004018064	0.010709505	0.014725569	-0.016064257	0.046854083	0.030789826	0.02811245	0.078982597	0.057563588	0.136546185
2	Boa	192	0.104166667	0.130208333	0.234375	0.015625	0.010416667	0.026041667	-0.046875	0.078125	0.03125	0.020833333	0.109375	0.067708333	0.177083333
3	Boa	328	0.085365854	0.134146341	0.219512195	0.009146341	0.00304878	0.012195122	-0.033536585	0.057926829	0.024390244	0.045731707	0.033536585	0.088414634	0.12195122
4	Boa	399	0.142857143	0.09273183	0.235588972	0.032581454	0	0.032581454	0.01754386	0.037593985	0.055137845	0.007518797	0.147869674	0.047619048	0.195488722
5	Boo	486	0.047325103	0.100823045	0.148148148	0.00617284	0.008230453	0.014403292	-0.014403292	0.053497942	0.03909465	0.00617284	-0.002057613	0.12345679	0.121399177
6	Boo	276	0.043478261	0.126811594	0.170289855	0.018115942	0.003623188	0.02173913	-0.036231884	0.054347826	0.018115942	0.010869565	0.003623188	0.097826087	0.101449275
7	Boo	237	0.092827004	0.097046414	0.189873418	0.018877637	0.004219409	0.021097046	0.025316456	0.033755274	0.05907173	0.025316456	0.130801688	0.054852321	0.185654008
8	Boo	231	0.12987013	0.090909091	0.220779221	0.025974026	0	0.025974026	0.017318017	0.012987013	0.03030303	0.012987013	0.0995671	0.082251082	0.181818182
9	Boo	221	0.104072398	0.14479638	0.248868778	0.004524887	0.018099548	0.022624434	-0.018099548	0.085972851	0.067873303	0.027149321	0.076923077	0.122171948	0.190905023
10	Fil	411	0.082725061	0.141119221	0.223844282	0.00973236	0.01216545	0.02189781	-0.0243309	0.051094891	0.02676399	0.01216545	0.077858881	0.072992701	0.150851582
11	Fil	385	0.05974026	0.137662338	0.197402597	0.007792208	0.01038961	0.018181818	-0.025974026	0.051948052	0.025974026	0.012987013	-0.005194805	0.142857143	0.137662338
12	Fil	583	0.04974271	0.116638079	0.166380789	0.013722127	0.005145798	0.018867925	-0.003430532	0.032590051	0.02915952	0.012006861	0.012006861	0.130360206	0.142367067
13	Fil	445	0.112359551	0.094382022	0.206741573	0.004494382	0.008988764	0.013483146	0.008988764	0.026966292	0.035955056	0.011235955	0.038202247	0.101123595	0.139325843
14	Fil	533	-0.00750469	0.148217636	0.140712946	0.015009381	0	0.015009381	-0.009380863	0.03564726	0.026266417	0.00750469	0.041275797	0.069418386	0.110694184
15	Fil	474	0.092827004	0.075949367	0.168776371	0.002109705	0.006329114	0.008438819	-0.014767932	0.025316456	0.010548523	0.002109705	0.065400844	0.056960205	0.122362869
16	Spo	321	0.052959502	0.080996885	0.133956386	0.024922118	0	0.024922118	0.021806854	0.028037383	0.049844237	0.009345794	0.102803738	0.021806854	0.124610592
17	Spo	465	0.107526882	0.047311828	0.15483871	0.040860215	0.002150538	0.043010753	0.060215054	0.008451813	0.066666667	0.010752688	0.176344086	0.017204301	0.193548387
18	Spo	468	0.051282051	0.115384615	0.166666667	0.049145299	0.002136752	0.051282051	0.025641028	0.017094017	0.042735043	0.008547009	0.132478632	0.036324798	0.168803419
19	Spo	420	0.145238095	0.083333333	0.228571429	0.035714286	0.002380952	0.038095238	0.038095238	0.028571429	0.066666667	0.004761905	0.119047619	0.057142857	0.176190476
20	Spo	311	0.057877814	0.061093248	0.118971061	0.022508039	0	0.022508039	0.048231511	0.006430888	0.054662379	0.012861736	0.106109325	0.025723473	0.131832797

Appendix C

Task 6 Network Centrality Measures for Documents

```
> print(stats)
```

	degree	betweenness	closeness	eigenvector
Board Game Review 1.txt	19	0.0	0.004048583	1.0000000
Board Game Review 2.txt	19	0.0	0.004901961	0.8379064
Board Game Review 3.txt	19	0.0	0.004784689	0.8571262
Board Game Review 4.txt	19	0.0	0.004255319	0.9546295
Book Review 1.txt	19	0.0	0.005649718	0.7295729
Book Review 2.txt	19	0.0	0.004784689	0.8580900
Book Review 3.txt	19	0.0	0.005181347	0.7948644
Book Review 4.txt	19	0.0	0.005681818	0.7285363
Book Review 5.txt	19	0.0	0.005524862	0.7477934
Film Review 1.txt	19	0.0	0.005376344	0.7669199
Film Review 2.txt	19	1.5	0.007246377	0.5757476
Film Review 3.txt	19	0.0	0.004255319	0.9547403
Film Review 4.txt	19	0.0	0.004629630	0.8829968
Film Review 5.txt	19	0.0	0.006329114	0.6563197
Film Review 6.txt	19	0.0	0.005347594	0.7697289
Sports Climbing News 1.txt	19	0.0	0.004716981	0.8685629
Sports Climbing News 2.txt	19	0.0	0.004901961	0.8370067
Sports Climbing News 3.txt	19	0.0	0.005102041	0.8085106
Sports Climbing News 4.txt	19	0.0	0.004329004	0.9402908
Sports Climbing News 5.txt	19	0.0	0.006097561	0.6807139

Appendix D

Task 7 Network Centrality Measures for Tokens

```
> print(stats_token)
```

	degree	betweenness	closeness	eigenvector
also	19	0	0.004587156	0.9173062
can	19	0	0.004219409	0.9871138
end	19	0	0.005586592	0.7583253
first	19	0	0.004273504	0.9738972
get	19	0	0.004975124	0.8463451
just	19	0	0.005494505	0.7696246
like	19	0	0.004854369	0.8675388
made	19	0	0.005208333	0.8145296
make	19	0	0.004629630	0.9088714
one	19	0	0.004166667	1.0000000
review	19	0	0.006097561	0.6969206
thing	19	0	0.005555556	0.7592578
time	19	0	0.004329004	0.9647623
turn	19	0	0.006060606	0.6980473
two	19	0	0.004694836	0.8930375
want	19	0	0.005128205	0.8261331
way	19	0	0.005524862	0.7669205
will	19	0	0.004807692	0.8761267
even	19	0	0.006060606	0.6971119
year	19	0	0.006134969	0.6911888

Appendix E

R-Code for All Tasks

Task 2

```
rm(list = ls())
library(slam)
library(tm)
library(SnowballC)
library(proxy)
library(igraph)
library(igraphdata)

setwd("~/MONASH/SOIT YR 3 SEM 1/FIT3152/Assignment 3")
col_names = file.path(".", "Corpus")
dir(col_names)

docs = Corpus(DirSource((col_names)))
summary(docs)
```

Task 3

```
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, content_transformer(tolower))

remove_unknowns <- content_transformer(function(x, pattern)
gsub(pattern, " ", x))

docs <- tm_map(docs, remove_unknowns, "'s")
docs <- tm_map(docs, remove_unknowns, "-")
docs <- tm_map(docs, remove_unknowns, "-")
docs <- tm_map(docs, remove_unknowns, "'")
docs <- tm_map(docs, remove_unknowns, '"')
docs <- tm_map(docs, remove_unknowns, '`')
docs <- tm_map(docs, remove_unknowns, ' ')

docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, stripWhitespace)
docs <- tm_map(docs, stemDocument, language = "english")

dtm <- DocumentTermMatrix(docs)

dtms <- removeSparseTerms(dtm, 0.45)
inspect(dtms)

dtms = as.data.frame(as.matrix(dtms))
# write.csv(dtms, "dtm.csv")
```

Task 4

```
cosine_distmatrix = proxy::dist(dtms, method = "cosine")

fit_cosine = hclust(cosine_distmatrix, method = "ward.D")

plot(fit_cosine, hang = -1, main = "Cosine Distance Cluster
Dendrogram")

cosine_groups = cutree(fit_cosine, k = 4)
cosine_groups

topics = c("book review", "book review", "book review", "book
review", "book review",
           "BG review", "BG review", "BG review", "BG review",
           "film review", "film review", "film review", "film
review", "film review", "film review",
           "SC news", "SC news", "SC news", "SC news", "SC news")

cosine_cf <- table(topics, cosine_groups)
cosine_cf = cosine_cf[,c(2,1,3,4)]
cosine_cf

cosine_accuracy = sum(diag(cosine_cf)) / sum(cosine_cf)
cosine_accuracy
```

Task 5

```
library(SentimentAnalysis)

SentimentA = analyzeSentiment(docs)

namelist = as.data.frame(as.table(summary(docs)))
namelist = head(namelist, nrow(SentimentA)) # ignore repeats
namelist$ID = substr(namelist$Var1, 1, 3)
namelist = as.data.frame(namelist[,4])
colnames(namelist) = "Genre"

SentimentA = cbind(namelist, SentimentA)
write.csv(x = SentimentA, "Sentiments Genre Groups.csv")
pdf(file = "Sentiments Genre Groups.pdf", height = 5, width = 10)

par( mfrow= c(2,2) )
boxplot(WordCount ~ Genre, data = SentimentA, frame = TRUE)
boxplot(SentimentQDAP ~ Genre, data = SentimentA, frame = TRUE)
boxplot(PositivityQDAP ~ Genre, data = SentimentA, frame = TRUE)
boxplot(RatioUncertaintyLM ~ Genre, data = SentimentA, frame = TRUE)
dev.off()

boa = SentimentA[SentimentA$Genre == "Boa",15]
boo = SentimentA[SentimentA$Genre == "Boo",15]
```

```

fil = SentimentA[SentimentA$Genre == "Fil",15]
spo = SentimentA[SentimentA$Genre == "Spo",15]

val1 = t.test(boa, boo, alternative = "greater")
val2 = t.test(boa, fil, alternative = "greater")
val3 = t.test(boa, spo, alternative = "greater")
val4 = t.test(boo, fil, alternative = "greater")
val5 = t.test(boo, spo, alternative = "greater")
val6 = t.test(fil, spo, alternative = "greater")

data_name = c(val1$data.name, val2$data.name, val3$data.name,
val4$data.name, val5$data.name, val6$data.name)
p_vals = c(val1$p.value, val2$p.value, val3$p.value, val4$p.value,
val5$p.value, val6$p.value)

p_vals_df = data.frame(genre = data_name, p_values = p_vals)

p_vals_df[order(p_vals_df$p_values),]

boa = SentimentA[SentimentA$Genre == "Boa",13]
boo = SentimentA[SentimentA$Genre == "Boo",13]
fil = SentimentA[SentimentA$Genre == "Fil",13]
spo = SentimentA[SentimentA$Genre == "Spo",13]

val1 = t.test(boa, boo, alternative = "less")
val2 = t.test(boa, fil, alternative = "less")
val3 = t.test(boa, spo, alternative = "less")
val4 = t.test(boo, fil, alternative = "less")
val5 = t.test(boo, spo, alternative = "less")
val6 = t.test(fil, spo, alternative = "less")

data_name = c(val1$data.name, val2$data.name, val3$data.name,
val4$data.name, val5$data.name, val6$data.name)
p_vals = c(val1$p.value, val2$p.value, val3$p.value, val4$p.value,
val5$p.value, val6$p.value)

p_vals_df = data.frame(genre = data_name, p_values = p_vals)

p_vals_df[order(p_vals_df$p_values),]

boa = SentimentA[SentimentA$Genre == "Boa",12]

```

```

boo = SentimentA[SentimentA$Genre == "Boo",12]
fil = SentimentA[SentimentA$Genre == "Fil",12]
spo = SentimentA[SentimentA$Genre == "Spo",12]

val1 = t.test(boa, boo, alternative = "greater")
val2 = t.test(boa, fil, alternative = "greater")
val3 = t.test(boa, spo, alternative = "greater")
val4 = t.test(boo, fil, alternative = "greater")
val5 = t.test(boo, spo, alternative = "greater")
val6 = t.test(fil, spo, alternative = "greater")

data_name = c(val1$data.name, val2$data.name, val3$data.name,
val4$data.name, val5$data.name, val6$data.name)
p_vals = c(val1$p.value, val2$p.value, val3$p.value, val4$p.value,
val5$p.value, val6$p.value)

p_vals_df = data.frame(genre = data_name, p_values = p_vals)
p_vals_df[order(p_vals_df$p_values),]

```

Task 6

```

dtms_matrix = as.matrix(dtms)

dtms_matrix = as.matrix((dtms_matrix > 0) + 0)

ByAbsMatrix = dtms_matrix %*% t(dtms_matrix)

diag(ByAbsMatrix) = 0

ByAbs = graph_from_adjacency_matrix(ByAbsMatrix, mode = "undirected",
weighted = TRUE)

plot(ByAbs)

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

print(stats)
head(stats[order(-stats$betweenness),])

```

```

head(stats[order(-stats$closeness),])
head(stats[order(-stats$eigenvector),])
head(stats[order(-stats$degree),])

ceb = cluster_edge_betweenness(as.undirected(ByAbs))

cfg = cluster_fast_greedy(as.undirected(ByAbs))

V(ByAbs)$size <- closeness(ByAbs)*3000

E(ByAbs)$width <- E(ByAbs)$weight*0.15

g_ceb = plot(ceb,as.undirected(ByAbs),
             vertex.label=V(ByAbs)$role,
             main="Edge Betweenness")
g_cfg = plot(cfg,as.undirected(ByAbs),
             vertex.label=V(ByAbs)$role,
             main="Fast Greedy",
             layout=layout_fruchterman_reingold,
             main="fruchterman.reingold")

```

Task 7

```

dtmsx = as.matrix(dtms)

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

ByTokenMatrix = t(dtmsx) %*% dtmsx

diag(ByTokenMatrix) = 0

ByToken = graph_from_adjacency_matrix(ByTokenMatrix, mode =
"undirected", weighted = TRUE)

plot(ByToken)

d = as.table(degree(ByToken))
b = as.table(betweenness(ByToken))
c = as.table(closeness(ByToken))
e = as.table(evcent(ByToken)$vector)
stats_token = as.data.frame(rbind(d,b,c,e))
stats_token = as.data.frame(t(stats_token))

colnames(stats_token) = c("degree", "betweenness", "closeness",
"eigenvector")

print(stats_token)
head(stats_token[order(-stats_token$betweenness),])
head(stats_token[order(-stats_token$closeness),])
head(stats_token[order(-stats_token$eigenvector),])
head(stats_token[order(-stats_token$degree),])

```

```

ceb_token = cluster_edge_betweenness(as.undirected(ByToken))

cfg_token = cluster_fast_greedy(as.undirected(ByToken))

V(ByToken)$size <- closeness(ByToken)*3000

E(ByToken)$width <- E(ByToken)$weight*0.15

g_ceb =
plot(ceb_token,as.undirected(ByToken),vertex.label=V(ByToken)$role,
     main="Edge Betweenness For Tokens")
g_cfg =
plot(cfg_token,as.undirected(ByToken),vertex.label=V(ByToken)$role,
     main="Fast Greedy For Tokens",
     layout=layout.fruchterman.reingold,
     main="fruchterman.reingold")

```

Task 8

```

dtmsa = as.data.frame(dtms)
dtmsa$ABS = rownames(dtmsa)
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 ) } }

colnames(dtmsb) = c("weight", "abs", "token")
dtmsc = dtmsb[dtmsb$weight != 0,]

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

g <- graph_from_data_frame(dtmsc, directed=FALSE)
bipartite_mapping(g)

V(g)$type <- bipartite_mapping(g)$type
V(g)$color <- ifelse(V(g)$type, "yellow", "orange")
V(g)$shape <- ifelse(V(g)$type, "circle", "square")
E(g)$color <- "gray"
plot(g)

cfg_bi= cluster_fast_greedy(as.undirected(g))

V(g)$size <- closeness(g)*1500

E(g)$width <- as.numeric(dtmsc$weight)*0.25

g_cfg = plot(cfg_bi,as.undirected(g),vertex.label=V(g)$role,

```

```
main="Fast Greedy for Bipartite",  
layout=layout.fruchterman.reingold,  
main="fruchterman.reingold")
```