

Assignment 2

Smitesh Patil

2023-03-10

PART 1

Section 1: Set up the libraries and check the input data

```
#importing all the necessary libraries
library(igraph)
library(ggraph)
library(ggrepel)
library(kableExtra)
library(gt)
library(tidyr)
library(dplyr)

# reading the graph
g<- read_graph(file="./WordPairs.txt",format="pajek")
#removing directions
g<- as.undirected(g)
g<- simplify(g)

#reading the cue text file for setting the cue attribute for graph
cues <- read.table("./cue.txt", header = F, sep="\t", skip=4)

#setting the cue attribute
V(g)$cue<-cues[[1]]

#checking the diameter value
print(paste0("Diameter of the network: ", diameter(g, weights = NA)))
```

```
## [1] "Diameter of the network: 7"
```

```
check_cue_words <- function(target_node_name1, target_node_name2){
# test if the selected words are cue words

  if(V(g)[target_node_name1]$cue & V(g)[target_node_name2]$cue){
    cat("Both target words are cue words \n")
  }else{
```

```

    cat("Both target words are NOT cue words \n")
    cat(target_node_name1, "cue = ", as.logical(V(g)[target_node_name1]$cue ), "\n")
    cat(target_node_name2, "cue = ", as.logical(V(g)[target_node_name2]$cue ), "\n")
  }
}

```

Section 2:

Q1.1 Extract a word association sub-network around each pair of words by sampling the graph in the vicinity of each word. The sub-network for each pair of words should be between 100 and 200 words. The objective is to build a word association network around your pair of words that will capture all the semantic contexts your target words belong to – without bringing in irrelevant contexts.

Q1.2 Once you have created the association network, select three authority/centrality measures and report the five words with highest values for each measure for each of your three networks. This must be shown in a table. Comment on the words in your table.

Q1.3 Produce a visualisation of the association network.

```

#
# UTILITY FUNCTIONS for
# 1. creating word association network,
# 2. centralities
# 4. plotting the graph
#in this cell

#-----Q1.1-----
#random walk function
random_walk_topic_network <- function(g,target_node_names, steps, walks, mode, topn){
  #initialising a empty vector
  vertices <- c()
  #looping on the two target names
  for (i in 1:2){
    #looping on the number of walks
    for (j in 1:walks){
      #running random walk and appending the vertices to the list
      vertices <- c(vertices, list(random_walk(g, target_node_names[i], steps, mode = mode)))
    }
  }

  #calculating the frequency of words appeared in the vertices vector and subsetting top n
  frequency_target <- head(sort(table(names(unlist(vertices)))), decreasing = TRUE), topn)
  #getting unique words
  unique_words <- names(frequency_target)

  #preparing and returning vertices in graph format
  vertices_in_word_association <- V(g)[name %in% unique_words]
  return(vertices_in_word_association)
}

#page rank function
page_rank_network <- function(g, target_node_names, topn, damping){
  #storing probabilities for all the vertices in the graph
  teleport_probs <- rep(0,vcount(g))
  teleport_probs[as.numeric(V(g)[target_node_names])] <- 1/length(target_node_names)
  #running page rank with parameters passed

```

```

pr <- page_rank(g, directed = F, personalized=teleport_probs, damping = damping)$vector
# getting top n vertices
top_n_pr <- order(pr, decreasing=TRUE)[1:topn]
top_n_pr<-V(g)[top_n_pr]

#returning the vertices
return(top_n_pr)
}

#-----Q1.2-----
#centrailites function for getting top vertices based on centrailites
centralities = function(word_association_network){
  #page rank centralities
  page_rank <- page_rank(word_association_network)$vector
  page_rank <- na.omit(page_rank[!names(page_rank) %in% c(target_word1, target_word2)])
  #getting five highest values
  page_rank <- sort(page_rank, decreasing = TRUE)[1:5]

  #betweenness centralities
  betweenness <- betweenness(word_association_network)
  betweenness <- betweenness[!names(betweenness) %in% c(target_word1, target_word2)]
  #getting five highest values
  betweenness <- sort(betweenness, decreasing = TRUE)[1:5]

  #eigen centralities
  eigen_centrality <- eigen_centrality(word_association_network)$vector
  eigen_centrality <- eigen_centrality[!names(eigen_centrality) %in% c(target_word1, target_word2)]
  #getting five highest values
  eigen_centrality <- sort(eigen_centrality, decreasing = TRUE)[1:5]

  #returning all the centralities in the list
  return(tibble("Page Rank" = names(page_rank) ,
                "Betweenness" = names(betweenness),
                "Eigen Centrality"= names(eigen_centrality)))
}

#-----Q1.3-----
#plotting the graph
plot_graph = function(association_network, vertex_size, label_size,
                      target_word1, target_word2, count){
  # getting top 25 nodes with high degree to label
  label_data <- names(sort(degree(association_network), decreasing = TRUE)[1:25])

  #setting vertex size
  vertex_size <- 2.5 + degree(association_network, mode = "all")/vertex_size
  cex_size <-2 + degree(association_network, mode = "all")/label_size

  #graph creation
  ggraph(association_network, layout = "fr")+
  #creating edges
  geom_edge_link(start_cap = circle(2.5, "mm"),
                end_cap = circle(2.5, "mm"),
                edge_width = 0.2,
                alpha = 0.2)+
  #parameters for nodes

```

```

geom_node_point(aes(size = vertex_size),
               alpha = 0.8,
               repel = TRUE,
               # setting distinct color for word pair
               colour = ifelse(V(association_network)$name %in%
                              c(target_word1, target_word2), "yellow", "red"))+
geom_node_text(
  # setting labels for word pair and top 25 nodes by degree
  aes(label = ifelse(V(association_network)$name %in% c(label_data, target_word1, target_word2),
                    name, NA)),
  fontface = "bold",
  position = "identity",
  size = cex_size,
  repel = TRUE

) +
#title
ggtitle(paste0("Figure No. ", count, " association network for words ", target_word1,
              " and ", target_word2))+
#turning off legend
guides(size = FALSE)+
theme(plot.title = element_text(size = 20,
                                hjust = 0.5))
}

```

Section 3: Simulate the function on three word pairs and justify the paramters

```

#target words
target_word1 <- "COURT"
target_word2 <- "LAWYER"

#checking the cue words
check_cue_words(target_word1, target_word2)

## Both target words are cue words

#calling the function
out <- random_walk_topic_network(g, c(target_word1, target_word2), 3, 40, "all", 160)

#creating a subgraph with the vertices returned by random walk/ Personalised page rank
word_association_network1 <- induced.subgraph(g, out)

#target words
target_word1 <- "CHILDREN"
target_word2 <- "PARENTS"

#checking the cue words
check_cue_words(target_word1, target_word2)

## Both target words are cue words

```

```

#calling the function
out <- page_rank_network(g, c(target_word1, target_word2), 140, 0.85)

#creating a subgraph with the vertices returned by random walk/ Personalised page rank
word_association_network2 <- induced.subgraph(g, out)

#target words
target_word1 <- "KING"
target_word2 <- "ROYAL"

#checking the cue words
check_cue_words(target_word1, target_word2)

```

```

## Both target words are cue words

```

```

#calling the function
out <- page_rank_network(g, c(target_word1, target_word2), 150, 0.95)

#creating a subgraph with the vertices returned by random walk/ Personalised page rank
word_association_network3 <- induced.subgraph(g, out)

#calling the centralities function
centrality1 = centralities(word_association_network1)
centrality2 = centralities(word_association_network2)
centrality3 = centralities(word_association_network3)

#plot dataframe
centrality <- tibble(tibble("No."=1:5), centrality1, centrality2, centrality3, .name_repair = "unique")
centrality %>%
  knitr::kable(
    format = "latex",
    align = "c",
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    caption = paste0("Centrality Score for three word pairs")
  ) %>%
  add_header_above(c("Index" = 1,
                     "Scores for wordpair Court & Lawyers" = 3,
                     "Scores for wordpair Children & Parents" = 3,
                     "Scores for wordpair King & Royal" = 3)) %>%
  kableExtra::kable_paper(
    position = "center",
    latex_options = c("striped", "repeat_header", "scale_down"),
    stripe_color = "gray!15",
    font_size = 8
  )

```

Table 1: Centrality Score for three word pairs

| Index | Scores for wordpair Court & Lawyers | | | Scores for wordpair Children & Parents | | | Scores for wordpair King & Royal | | |
|-------|-------------------------------------|-------------|------------------|--|-------------|------------------|----------------------------------|-------------|------------------|
| No. | Page Rank | Betweenness | Eigen Centrality | Page Rank | Betweenness | Eigen Centrality | Page Rank | Betweenness | Eigen Centrality |
| 1 | COURT | COURT | LAWYER | CHILDREN | PARENTS | DAD | QUEEN | POWER | QUEEN |
| 2 | LAWYER | LAWYER | COURT | CHILD | CHILDREN | MOM | BLUE | HOUSE | THRONE |
| 3 | LAW | CLAIM | JUDGE | OLD | KIDS | MOTHER | HOUSE | LOVE | ROYALTY |
| 4 | JUDGE | CASE | LAW | PARENTS | SCHOOL | FATHER | RICH | GIRL | CROWN |
| 5 | MONEY | CRIMINAL | ATTORNEY | BAD | FRIEND | OLD | DOG | EMPIRE | EMPEROR |

Section 3.1: Analysis of network and effects of Personalised Page Rank and Random walk on association network creation.

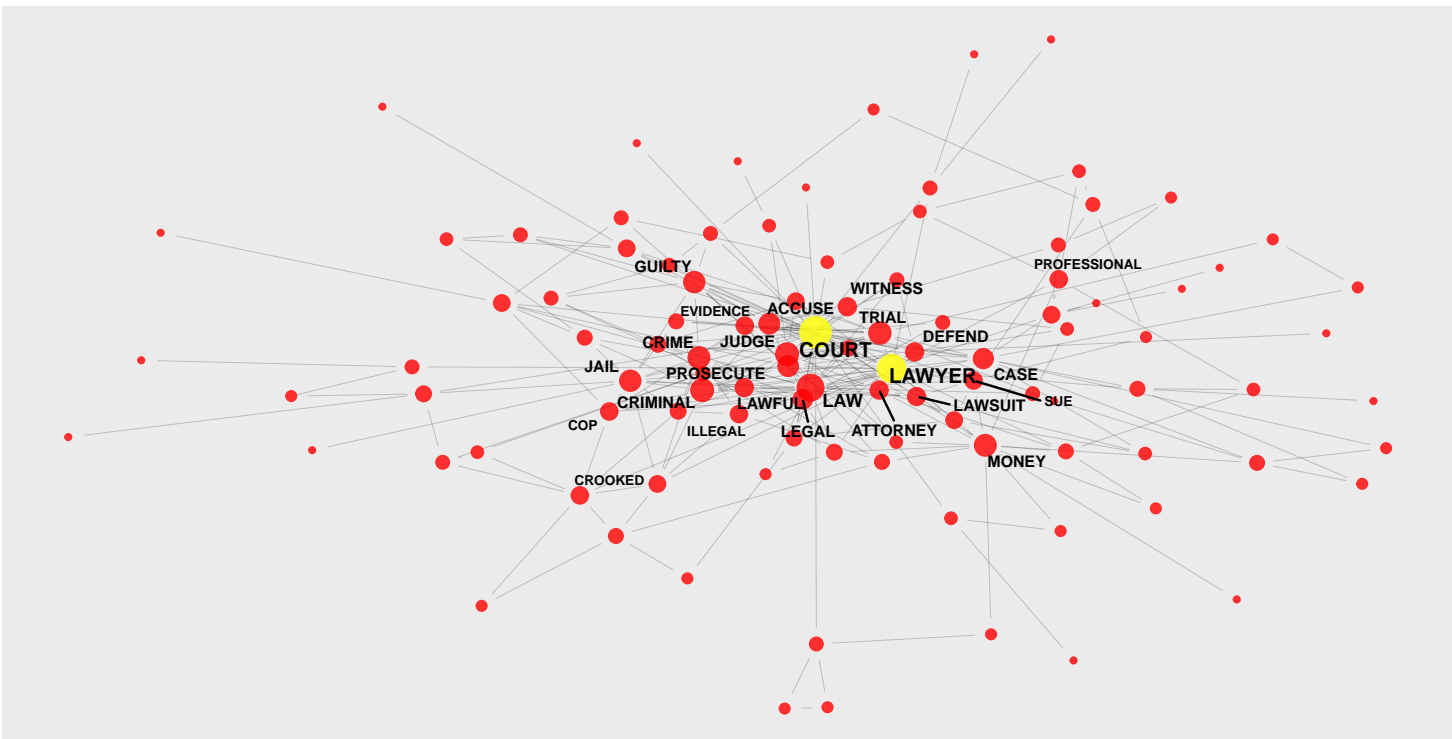
Three word pairs choose to create word network, they are “Court and Lawyer”, “Children and Parents”, and “King and Royal”. For creating the word association network, random walk algorithm was used for “Court and Lawyer” pair, and personalised pagerank algorithm was used for the other two pairs, The reason for choosing these specific algorithms were that they were optimal in creating word association network that would generate communities that were relatively coherent compared to others.

For Random Walk it was observed that as the number of walks and size of vertices increases, it increases the size of communities formed later during community detection and hence the number of walks for the one pair is restricted to 40 walks and 160 vertices.

For Personalised Page Rank every time nodes are ranked randomly some nodes are given more weight so that there is a chance for other nodes to be more relevant based on the damping factor, it was found that a high damping factor yields better results on community detection, i.e more communities of ideal size and coherent nodes.

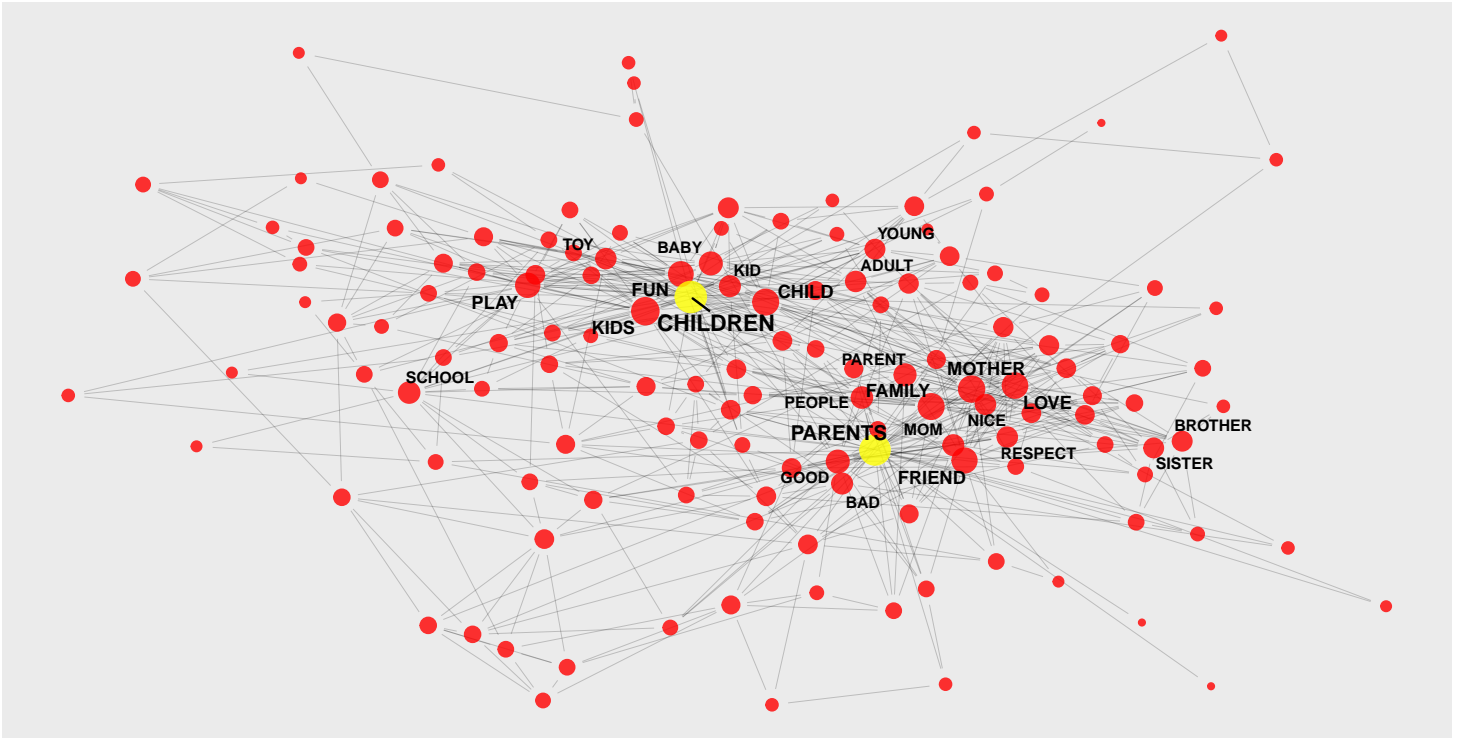
```
#plotting network
plot_graph(word_association_network1, 10, 30, "COURT", "LAWYER", 1)
```

Figure No. 1 association network for words COURT and LAWYER



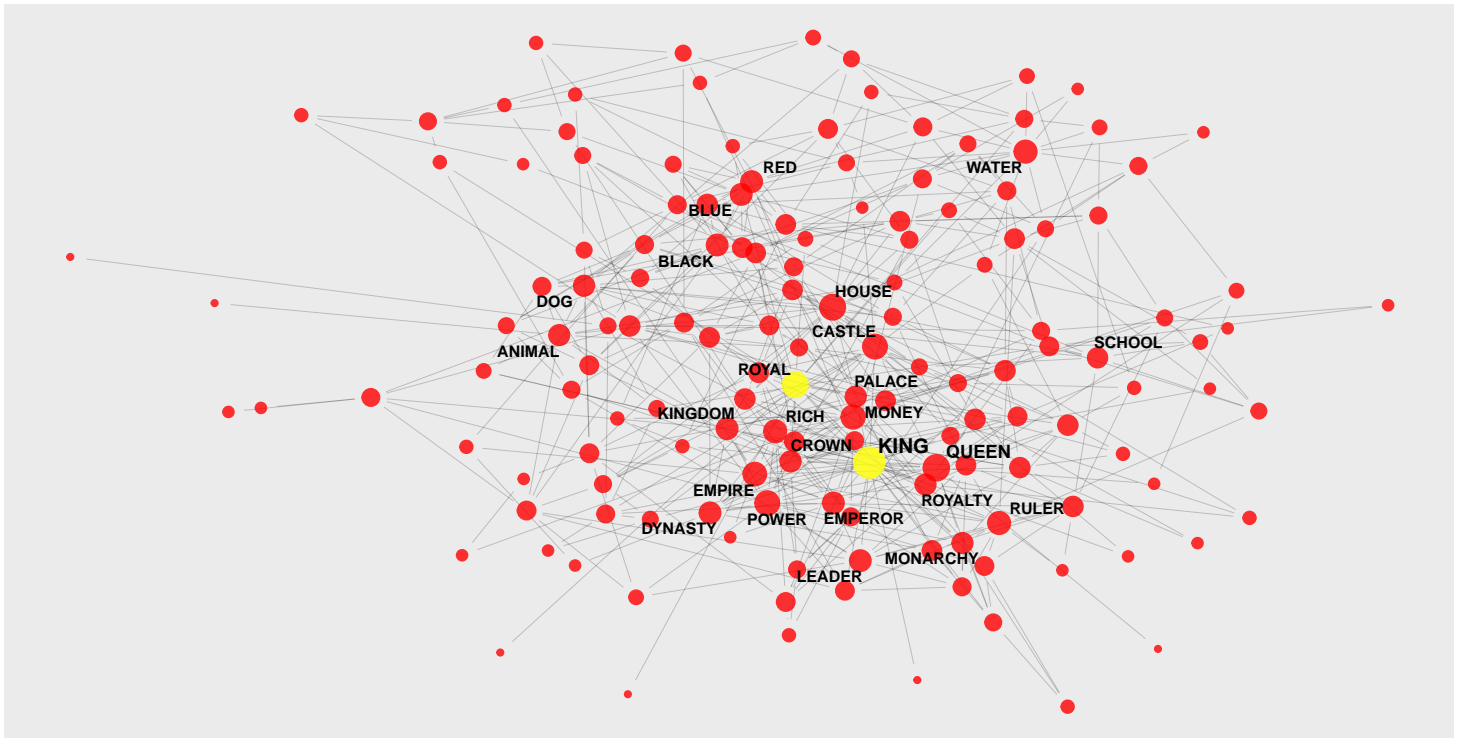
```
#plotting network
plot_graph(word_association_network2, 10, 30, "CHILDREN", "PARENTS", 2)
```

Figure No. 2 association network for words CHILDREN and PARENTS



```
#plotting the graph
plot_graph(word_association_network3, 10, 30, "KING", "ROYAL", 3)
```

Figure No. 3 association network for words KING and ROYAL



PART 2

Section 1: Detect and display community tables

Q2.1 Using an appropriate community detection algorithm, determine the communities in each of the three word association networks you created in Part 1. You are expected to evaluate different community detection algorithms and you should justify the one you have selected.

Q2.2 Produce a table listing the words in each community. You should do this using a table formatting package such as Kable.

```
#creating optimal communities for word association network based on trial and error

cluster1 <- cluster_fluid_communities(word_association_network1, no.of.communities = 13)

cluster2 <- cluster_label_prop(word_association_network2)

cluster3 <- cluster_edge_betweenness(word_association_network3)

#### UTILITY FUNCTIONS FOR TASK 2
# create community table
# print community table with size

#create community table with size
create_community_table <- function(clustering_data){
  strings <- c()
  lengths <- c()
  #loop over each community
  for(i in 1:length(clustering_data)){
    #append the clustering data into a string
    string = ""
```



```

for(word in clustering_data[[i]]){
  string = paste0(string, word, " , ")
}

#store the string and the size
strings <- append(strings, string)
lengths <- append(lengths, length(clustering_data[[i]]))
}

#create a tibble
df <- tibble(strings, lengths) %>%
  filter(lengths > 1 )
colnames(df) <- c("Cluster","Size")

#return the tibble
return(df)
}

# print then df created in create_community_table
print_community_table <- function(community, target_word1, target_word2, cluster){
  #preprocess table
  community %>%
  #arrange by size for printing
  arrange(desc(Size)) %>%
  # create index
  mutate(No = 1:nrow(community)) %>%
  #order the table for printing
  select(No, Cluster, Size) %>%
  #discarding too small clusters
  filter(Size > 3) %>%
  # kable printing with params
  knitr::kable(
    format = "latex",
    align = "l",
    booktabs = TRUE,
    longtable = TRUE,
    linesep = "",
    caption = paste0("Communities detected for word pair ", target_word1, " and ",
                     target_word2, "Modularity ", round(modularity(cluster), 2))
  ) %>%
  column_spec(column = 2, width = "7in") %>%
  kableExtra::kable_paper(
    position = "left",
    latex_options = c("striped", "repeat_header"),
    stripe_color = "gray!15"
  )
}

#create and print the tables for the task
community_table1 <- create_community_table(cluster1)
print_community_table(community_table1, "Court", "Lawyer", cluster1)

```

Table 2: Communities detected for word pair Court and Lawyer-
Modularity 0.32

| No | Cluster | Size |
|----|---|------|
| 1 | ACCUSE , GUILT , PROSECUTE , COURT , JUDGE , TRIAL , SIN , SCARE , WITNESS , ERROR , CONFESSION , JUDGMENT , EXAMPLE , TESTIFY , | 14 |
| 2 | DEFEND , HOUSE , ADJOURN , END , ARMY , SPEAKER , BUILDER , STAND , RIGHTS , PEOPLE'S , JESTER , MASON , | 12 |
| 3 | CONCRETE , PEOPLE , STREET , DIVORCE , ROAD , CROOKED , POLITICIAN , GROUNDS , BREAK-UP , GAVEL , MAYOR , | 11 |
| 4 | CRIME , LAWYER , CRIMINAL , BRIBE , ILLEGAL , CORRUPT , LEGAL , LAWS , UNLAWFUL , LAWFUL , PERJURY , | 11 |
| 5 | TOP , MONEY , ACCEPT , CHARGE , MOUNTAIN , SUMMIT , RICH , ATTORNEY , BUTLER , INVENT , | 10 |
| 6 | CONTRACT , LAW , JAIL , COP , POLICY , BAIT , PUNISHMENT , JUSTIFY , OATH , | 9 |
| 7 | BOOK , CLAIM , FIRM , SITUATION , CASE , FILE , SUBJECT , EVIDENCE , | 8 |
| 8 | SMART , EXPERT , PROFESSIONAL , DOCTOR , HEAL , NURSE , VET , | 7 |
| 9 | SPORT , TENNIS , TROUBLE , FAULT , FLUNK , | 5 |
| 10 | ME , ATHLETIC , GIRL , SUE , LAWSUIT , | 5 |
| 11 | GUILTY , HONEST , PLEASE , PLEAD , INNOCENT , | 5 |
| 12 | NOTE , EXECUTIVE , BRIEFCASE , ATTACHE , | 4 |

```
community_table2 <- create_community_table(cluster2)
print_community_table(community_table2, "Children", "Parent", cluster2)
```

Table 3: Communities detected for word pair Children and Parent-
Modularity 0.64

| No | Cluster | Size |
|----|--|------|
| 1 | UNDERSTAND , SMART , YOUNG , MATURE , OLD , NEW , DEVELOP , GRANDMA , GRANDPARENTS , | 9 |
| 2 | CHILD , RESPECT , ADULT , KID , PARENT , AUTHORITY , RESPONSIBILITY , DEVELOPMENT , | 8 |
| 3 | GOOD , BAD , INFLUENCE , OKAY , NICE , APPROVAL , PERSUADE , MORALS , | 8 |
| 4 | STOP , ALLOW , ASK , NO , LET , PERMIT , PERMISSION , FORBID , | 8 |
| 5 | ADULTS , PARENTS , GROWN-UPS , GENE , OVERPOWER , GUARDIANS , FOLKS , | 7 |
| 6 | WORK , SCHOOL , MONEY , BOOK , TEACH , HORSE , | 6 |
| 7 | KIDS , CHILDREN , RED , TOYS , CLAY , PLAY DOUGH , | 6 |
| 8 | WATER , TREE , PARK , SWING , SLIDE , PLAYGROUND , | 6 |
| 9 | PLAY , GAME , FUN , GAMES , BALL , PLAYING , | 6 |
| 10 | FRIEND , FAMILY , RELATIVE , FRIENDS , DISOWN , | 5 |
| 11 | MUSIC , LOUD , NOISE , NOISY , | 4 |
| 12 | HOME , HOUSE , DOG , NEIGHBORHOOD , | 4 |
| 13 | BABY , INFANT , NURSERY , RHYME , | 4 |
| 14 | MOTHER , FATHER , CRITICAL , DAUGHTER , | 4 |
| 15 | RESTRICTION , LAW , RULES , OBEY , | 4 |

```
community_table3 <- create_community_table(cluster3)
print_community_table(community_table3, "King", "Royal", cluster3)
```

Table 4: Communities detected for word pair King and RoyalMod-
ularity 0.36

| No | Cluster | Size |
|----|--|------|
| 1 | BLACK , PAPER , TREE , GREEN , HOUSE , BLUE , FIRE , CLEAN , BIRD , RED , CAT , LIGHT , WHITE , DOG , HAIR , COLOR , FLOWER , PURPLE , BUTTERFLY , | 19 |
| 2 | ONE , COURT , KING , GOVERNMENT , MEASURE , ANARCHY , RULER , MONARCHY , RULE , PRESIDENT , LEADER , DICTATOR , MONARCH , OLIGARCHY , TOPPING , | 15 |
| 3 | GOOD , FOOD , FIGHT , HAPPY , LOVE , ARMY , DRINK , GOD , SMELL , CHURCH , RELIGION , FRANCE , | 12 |
| 4 | POWER , MONEY , BOOK , WEALTH , TELEVISION , JEWEL , RING , RICH , GOLD , DIAMOND , DYNASTY , | 11 |
| 5 | STRONG , PEOPLE , BAD , MAN , CITY , OLD , BIG , FRIEND , POOR , MUMMY , | 10 |
| 6 | WATER , RAIN , GAME , FUN , HOT , SUN , BALL , BEACH , LONDON , FLUSH , | 10 |
| 7 | PAIN , SICK , HURT , FALL , HEAD , DEATH , SAD , GUN , COLD , | 9 |
| 8 | ME , BEAUTIFUL , BEE , QUEEN , BRIDE , PRINCESS , PRINCE , ROYALTY , | 8 |
| 9 | MUSIC , COUNTRY , CLOTHES , CHINA , ENGLAND , ROMAN , EMPEROR , | 7 |
| 10 | WORK , MATH , HARD , HORSE , ARMOR , KNIGHT , MEDIEVAL , | 7 |
| 11 | HIGH , SKY , SCHOOL , MOVIE , BUILDING , EMPIRE , | 6 |
| 12 | HOME , SAND , MANSION , CASTLE , MOAT , PALACE , | 6 |
| 13 | ANIMAL , LION , FROG , KINGDOM , SNAKE , | 5 |
| 14 | TOP , CROWN , HAT , TOILET , THRONE , | 5 |
| 15 | CAR , SHIP , BOAT , FISH , DRAG , | 5 |
| 16 | GIRL , BEAUTY , BABY , SMALL , SLEEP , | 5 |
| 17 | FAMILY , ROYAL , PEASANT , REGAL , | 4 |

Q2.3 For each community determine a one word label for the community. You will do this by treating each community as a new subgraph and calculating the most central node in the subgraph. This will be the community's label. The label for each community should be shown in the same table where the communities are listed.

UTILITY FUNCTIONS

```
# 1. create tibble with community label
# 2. plot the tibble with community labels

# function to create df with communities with label
define_community_labels <- function(cluster, dataframe){
  community_label = c()
  #looping on clusters
  for (i in 1:length(cluster)){
    # condition on length of cluster
    if (length(cluster[[i]]) > 1){
      #subsetting vertices
      vertices_in_community <- V(g)[name %in% cluster[[i]]]
      #create subgraph
      community_graph <- induced.subgraph(g, vertices_in_community)
      #calculating centrality
      page_rank <- page_rank(community_graph)$vector
      #highest ranked node as label
      label = sort(page_rank, decreasing = TRUE)[1]
      community_label <- c(community_label, label)
    }
  }

  #adding the community label by mutating
  dataframe <- dataframe %>% mutate(community_label = names(community_label))
  return(dataframe)
}
```

```

}

# function to plot dataframe
plot_dataframe = function(labelled_data, target_word1, target_word2, cluster){
  labelled_data %>%
    #preprocess
    arrange(desc(Size)) %>%
    #add index
    mutate(No = 1:nrow(labelled_data)) %>%
    #discarding small communities
    filter(Size > 3) %>%
    #arranging the dataframe
    select(No, Cluster, community_label, Size) %>%
    #tables print
    knitr::kable(
      format = "latex",
      align = "l",
      booktabs = TRUE,
      longtable = TRUE,
      linesep = "",
      #printing the modularilty and word pairs
      caption = paste0("Communities detected for word pair ", target_word1, " and ",
                        target_word2, " Modularity ", round(modularity(cluster), 2))
    ) %>%
    column_spec(column = 2, width = "6in") %>%
    kableExtra::kable_paper(
      position = "left",
      latex_options = c("striped", "repeat_header", "scale_down"),
      stripe_color = "gray!15"
    )
}

#calling functions and plotting tables

clus1_comm_label<-define_community_labels(cluster1, community_table1)
plot_dataframe(clus1_comm_label, "Court", "Lawyer", cluster1)

```

Table 5: Communities detected for word pair Court and Lawyer
Modularity 0.32

| No | Cluster | community_label | Size |
|----|---|-----------------|------|
| 1 | ACCUSE , GUILT , PROSECUTE , COURT , JUDGE , TRIAL , SIN , SCARE , WITNESS , ERROR , CONFESSION , JUDGMENT , EXAMPLE , TESTIFY , | COURT | 14 |
| 2 | DEFEND , HOUSE , ADJOURN , END , ARMY , SPEAKER , BUILDER , STAND , RIGHTS , PEOPLE'S , JESTER , MASON , | DEFEND | 12 |
| 3 | CONCRETE , PEOPLE , STREET , DIVORCE , ROAD , CROOKED , POLITICIAN , GROUNDS , BREAK-UP , GAVEL , MAYOR , | ROAD | 11 |
| 4 | CRIME , LAWYER , CRIMINAL , BRIBE , ILLEGAL , CORRUPT , LEGAL , LAWS , UNLAWFUL , LAWFUL , PERJURY , | ILLEGAL | 11 |
| 5 | TOP , MONEY , ACCEPT , CHARGE , MOUNTAIN , SUMMIT , RICH , ATTORNEY , BUTLER , INVENT , | MONEY | 10 |
| 6 | CONTRACT , LAW , JAIL , COP , POLICY , BAIT , PUNISHMENT , JUSTIFY , OATH , | LAW | 9 |
| 7 | BOOK , CLAIM , FIRM , SITUATION , CASE , FILE , SUBJECT , EVIDENCE , | CASE | 8 |
| 8 | SMART , EXPERT , PROFESSIONAL , DOCTOR , HEAL , NURSE , VET , | DOCTOR | 7 |

| | | | |
|----|---|-----------|---|
| 9 | SPORT , TENNIS , TROUBLE , FAULT , FLUNK , | TENNIS | 5 |
| 10 | ME , ATHLETIC , GIRL , SUE , LAWSUIT , | SUE | 5 |
| 11 | GUILTY , HONEST , PLEASE , PLEAD , INNOCENT , | GUILTY | 5 |
| 12 | NOTE , EXECUTIVE , BRIEFCASE , ATTACHE , | BRIEFCASE | 4 |

```
clus2_comm_label<-define_community_labels(cluster2, community_table2)
plot_dataframe(clus2_comm_label, "Children", "Parent", cluster2)
```

Table 6: Communities detected for word pair Children and Parent
Modularity 0.64

| No | Cluster | community_label | Size |
|----|--|-----------------|------|
| 1 | UNDERSTAND , SMART , YOUNG , MATURE , OLD , NEW , DEVELOP , GRANDMA , GRANDPARENTS , | OLD | 9 |
| 2 | CHILD , RESPECT , ADULT , KID , PARENT , AUTHORITY , RESPONSIBILITY , DEVELOPMENT , | CHILD | 8 |
| 3 | GOOD , BAD , INFLUENCE , OKAY , NICE , APPROVAL , PERSUADE , MORALS , | GOOD | 8 |
| 4 | STOP , ALLOW , ASK , NO , LET , PERMIT , PERMISSION , FORBID , | ALLOW | 8 |
| 5 | ADULTS , PARENTS , GROWN-UPS , GENE , OVERPOWER , GUARDIANS , FOLKS , | ADULTS | 7 |
| 6 | WORK , SCHOOL , MONEY , BOOK , TEACH , HORSE , | SCHOOL | 6 |
| 7 | KIDS , CHILDREN , RED , TOYS , CLAY , PLAY DOUGH , | CHILDREN | 6 |
| 8 | WATER , TREE , PARK , SWING , SLIDE , PLAYGROUND , | PLAYGROUND | 6 |
| 9 | PLAY , GAME , FUN , GAMES , BALL , PLAYING , | FUN | 6 |
| 10 | FRIEND , FAMILY , RELATIVE , FRIENDS , DISOWN , | FAMILY | 5 |
| 11 | MUSIC , LOUD , NOISE , NOISY , | LOUD | 4 |
| 12 | HOME , HOUSE , DOG , NEIGHBORHOOD , | HOUSE | 4 |
| 13 | BABY , INFANT , NURSERY , RHYME , | BABY | 4 |
| 14 | MOTHER , FATHER , CRITICAL , DAUGHTER , | MOTHER | 4 |
| 15 | RESTRICTION , LAW , RULES , OBEY , | RULES | 4 |

```
clus3_comm_label<-define_community_labels(cluster3, community_table3)
plot_dataframe(clus3_comm_label, "King", "Royal", cluster3)
```

Table 7: Communities detected for word pair King and Royal Modularity 0.36

| No | Cluster | community_label | Size |
|----|--|-----------------|------|
| 1 | BLACK , PAPER , TREE , GREEN , HOUSE , BLUE , FIRE , CLEAN , BIRD , RED , CAT , LIGHT , WHITE , DOG , HAIR , COLOR , FLOWER , PURPLE , BUTTERFLY , | WHITE | 19 |
| 2 | ONE , COURT , KING , GOVERNMENT , MEASURE , ANARCHY , RULER , MONARCHY , RULE , PRESIDENT , LEADER , DICTATOR , MONARCH , OLIGARCHY , TOPPING , | RULER | 15 |
| 3 | GOOD , FOOD , FIGHT , HAPPY , LOVE , ARMY , DRINK , GOD , SMELL , CHURCH , RELIGION , FRANCE , | GOD | 12 |
| 4 | POWER , MONEY , BOOK , WEALTH , TELEVISION , JEWEL , RING , RICH , GOLD , DIAMOND , DYNASTY , | MONEY | 11 |
| 5 | STRONG , PEOPLE , BAD , MAN , CITY , OLD , BIG , FRIEND , POOR , MUMMY , | PEOPLE | 10 |
| 6 | WATER , RAIN , GAME , FUN , HOT , SUN , BALL , BEACH , LONDON , FLUSH , | SUN | 10 |
| 7 | PAIN , SICK , HURT , FALL , HEAD , DEATH , SAD , GUN , COLD , | HURT | 9 |
| 8 | ME , BEAUTIFUL , BEE , QUEEN , BRIDE , PRINCESS , PRINCE , ROYALTY , | PRINCESS | 8 |
| 9 | MUSIC , COUNTRY , CLOTHES , CHINA , ENGLAND , ROMAN , EMPEROR , | COUNTRY | 7 |
| 10 | WORK , MATH , HARD , HORSE , ARMOR , KNIGHT , MEDIEVAL , | KNIGHT | 7 |
| 11 | HIGH , SKY , SCHOOL , MOVIE , BUILDING , EMPIRE , | BUILDING | 6 |

| | | | |
|----|--|--------|---|
| 12 | HOME , SAND , MANSION , CASTLE , MOAT , PALACE , | CASTLE | 6 |
| 13 | ANIMAL , LION , FROG , KINGDOM , SNAKE , | ANIMAL | 5 |
| 14 | TOP , CROWN , HAT , TOILET , THRONE , | CROWN | 5 |
| 15 | CAR , SHIP , BOAT , FISH , DRAG , | BOAT | 5 |
| 16 | GIRL , BEAUTY , BABY , SMALL , SLEEP , | BEAUTY | 5 |
| 17 | FAMILY , ROYAL , PEASANT , REGAL , | ROYAL | 4 |

#save the tables for analysis

```
write.csv(clus1_comm_label, "table1.csv")
write.csv(clus2_comm_label, "table2.csv")
write.csv(clus3_comm_label, "table3.csv")
```