Web&Network

2023-03-24

Mean Degree of Cue Words

```
library(igraph)
library(ggraph)

#Read File
g<- read_graph(file="./WordPairs.txt",format="pajek")

# the graph should be undirected
g<- as.undirected(g)

# Simple graphs are graphs which do not contain loop and multiple edges.
g<-simplify(g)

# read the cue.txt file into R
# skip first 4 lines which are header lines
cues <- read.table("./cue.txt", header = F, sep="\t", skip=4)

# add a cue value as vertex attribute
# each cue value is 0 or 1
V(g)$cue<-cues[[1]]

# The cue words are much better represented in the dataset
# In other words they have high degree - many nodes
# connect to them

deg<-degree(g)

cat("mean degree of cue words", mean(deg[which(V(g)$cue==1)]))
```

```
## mean degree of cue words 23.71642
```

Mean Degree of Non Cue Words

```
cat("mean degree of NON-cue words", mean(deg[which(V(g)$cue==0)]))
```

```
## mean degree of NON-cue words 1.527951
```

Check for Target Word Pairs being Cue Words

3 Pairs of Words are chosen from the WordPairs.txt file. All 3 pairs of words are checked if they are cue words or not with the help of cue.txt file. After recieving a positive result of 3 word pairs being cue words further processing is done to create a word-association subgraph of all 3 word pairs(ANIMALS,ZOO)(FOOD,MONEY)(RACE,WIN) are explained and implemented below.

```
#' select two word
#' these must be cue words
target_node_name1 <- "ANIMALS"
target_node_name2 <- "ZOO"

target_node_name3 <- "FOOD"
target_node_name4 <- "MONEY"

target_node_name5 <- "RACE"
target_node_name6 <- "WIN"
# 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")

}
```

```
## Both target words are cue words
```

```
if(V(g)[target_node_name3]$cue & V(g)[target_node_name4]$cue){
  cat("Both target words are cue words \n")
}else{
  cat("Both target words are NOT cue words \n")
  cat(target_node_name3, "cue = ", as.logical(V(g)[target_node_name3]$cue ),"\n")
  cat(target_node_name4, "cue = ", as.logical(V(g)[target_node_name4]$cue ),"\n")

}
```

```
## Both target words are cue words
```

```
if(V(g)[target_node_name5]$cue & V(g)[target_node_name6]$cue){
  cat("Both target words are cue words \n")
}else{
  cat("Both target words are NOT cue words \n")
  cat(target_node_name5, "cue = ", as.logical(V(g)[target_node_name5]$cue ),"\n")
  cat(target_node_name6, "cue = ", as.logical(V(g)[target_node_name6]$cue ),"\n")

}
```

```
## Both target words are cue words
```

Part 1

Q.1.1

Personalised Page Rank Function

The Personalised Page Rank Function is used to calculate a word association sub-network graph for 3 Pairs of Words. The parameters taken for implementing the function are undirected graph g, damping,target nodes(word pairs) and topn related words for sub network.

Value 110 is taken for topn. More the number of topn more the association graph created would be cluttered and relationships of nodes would not be comprehendable for the user. 110 provides a good visualization as well as decent amount of relatable and associated words of the word pair considered.

Damping factor considered is 0.85. High damping factor results in increase of chance for teleporting to the neighbour nodes.

Personalised Page Rank function is used as it simulates a participant and helps in associating the word the participant would most probably write in response after looking at the target word and provides all the authorative words from the perspective of that target word pairs.

```
library(igraph)

ppr_topic_network <- function(g,target_node_names, topn,damping = 0.85) {

#set the target node names chosen from Word Pairs.txt
topic<-target_node_names

# set initial vector of zeros
teleport_probab <- rep(0,vcount(g))

# we set the probability of returning to the topic node
teleport_probab[as.numeric(V(g)[topic])]<-1/length(topic)

#set directed to False since undirected graph used
page_r<- page_rank(g,  directed = F,
                  personalized=teleport_probab,
                  damping = 0.85)$vector

# the vertex ids of the top n scores
top_n_pr <- order(page_r, decreasing=TRUE)[1:topn]

#the top n vertices
top_n_pr<-V(g)[top_n_pr]

#Create subgraph of Word Pair / target_node_names
subgraph1 <- induced.subgraph(g,top_n_pr)

return(subgraph1)

}
```

Calling Personalised Page Rank Function

```
target_node_names<-c(target_node_name1,target_node_name2)

#parameters passed
topn<- 110
mode <-"all"
damping = 0.85

# call ppr function for finding a topic sub graph
# Calling Word Pair 1
topic_graph<- ppr_topic_network(g,target_node_names, topn,damping)

#Calling Word Pair 2
target_node_names2<-c(target_node_name3,target_node_name4)
topic_graph1<- ppr_topic_network(g,target_node_names2, topn,damping)

#Calling Word Pair 3
target_node_names3<-c(target_node_name5,target_node_name6)
topic_graph2<- ppr_topic_network(g,target_node_names3, topn,damping)
```

Q.1.2 Centrality Measures to report 5 words with highest values

The 3 Centrality measures suited best to report 5 words with highest values of 3 Word pairs are:

1) Betweenness Centrality - Betweeness Centrality will utilize the number of shortest path that belongs to Word Pair 1,2,3 to obtain highest values.

2) Page Rank Centrality - Page Rank Centrality provides all the important nodes connected to Word Pair 1,2,3 to obtain highest values.

3) Eigen Vector Centrality - In Eigen Vector Centrality the centrality a node gets from its network neighbours is proportional to their centrality divided by their out degree of the Word Pair 1,2,3 to obtain highest values.

The top 5 values differ as per the subnetwork size(topn) value passed in the function written above. The top 5 words formed from the centrality measures are related to the target words provided.

Betweeness Centrality for Word Pair 1: ANIMALS and ZOO

```
between<- betweenness(topic_graph, weights=NA)
# n is number of highest values
n <-5

# the vertex ids of the top n betweenness scores
topn <- order(between, decreasing=TRUE)[1:n]

# the top n vertices
topn <-names(V(topic_graph)[topn])
```

Table 1: Centrality for Word Pair 1: ANIMALS and ZOO

Betweeness	Page Rank	Eigen Vector
ANIMAL	ANIMAL	ANIMAL
ANIMALS	ANIMALS	CAT
DOG	DOG	DOG
CAT	CAT	BEAR
TREE	BEAR	LION

Page Rank Centrality for Word Pair 1: ANIMALS AND ZOO

```
pr<- page_rank(topic_graph,directed = TRUE, damping = 0.85,weights = NA)$vector

n <-5
# the vertex ids of the top n scores
top_n_pr <- order(pr, decreasing=TRUE)[1:n]

# the top n vertices
top_n_pr<-names(V(topic_graph)[top_n_pr])
```

Eigenvector Centrality for Word Pair 1: ANIMALS AND ZOO

```
eigen<- eigen_centrality(topic_graph, weights=NA)$vector
n <- 5
# the vertex ids of the top n scores
top_n_eigen <- order(eigen, decreasing=TRUE)[1:n]

# the top n vertices
top_n_eigen<-names(V(topic_graph)[top_n_eigen])
```

Table for Centralities of Word Pair 1: ANIMALS and ZOO

```
library(kableExtra)
wordpair1 <- data.frame(topn,top_n_pr,top_n_eigen)
names(wordpair1) <- c("Betweeness", "Page Rank", "Eigen Vector")
kable(wordpair1,caption = "Centrality for Word Pair 1: ANIMALS and ZOO",booktabs=F) %>%
kable_styling(latex_options = c("striped", "scale_down"))
```

Ans) For Word Pair 1: ANIMALS AND ZOO, the Betweeness Centrality,Eigen Vector Centrality and Page Rank Centrality gives top 5 values as mentioned in above code.It is observed that the Word 'ANIMAL','DOG' and 'CAT' are common for all centralities. Betweeness Centrality and Page Rank Centrality provide a similar output due to similar working of centrality calculations to find top 5 values as explained above.

Betweeness Centrality for Word Pair 2: FOOD and MONEY

```
between2<- betweenness(topic_graph1, weights=NA)
n <-5
# the vertex ids of the top n betweenness scores
topn2 <- order(between2, decreasing=TRUE)[1:n]

# the top n vertices
topn2 <-names(V(topic_graph1)[topn2])
```

Page Rank Centrality for Word Pair 2: FOOD and MONEY

```
pr2<- page_rank(topic_graph1,directed = TRUE, damping = 0.85,weights = NA)$vector

n <-5
# the vertex ids of the top n scores
top_n_pr2 <- order(pr2, decreasing=TRUE)[1:n]

# the top n vertices
top_n_pr2<-names(V(topic_graph1)[top_n_pr2])
```

Eigenvector Centrality for Word Pair 2: FOOD and MONEY

```
eigen2<- eigen_centrality(topic_graph1, weights=NA)$vector
n <- 5
# the vertex ids of the top n scores
top_n_eigen2 <- order(eigen2, decreasing=TRUE)[1:n]

# the top n vertices
top_n_eigen2<-names(V(topic_graph1)[top_n_eigen2])
```

Table for Centralities of Word Pair2: FOOD and MONEY

```
wordpair2 <- data.frame(topn2,top_n_pr2,top_n_eigen2)
names(wordpair2) <- c("Betweeness", "Page Rank", "Eigen Vector")
kable(wordpair2,caption = "Centrality for Word Pair 2: FOOD and MONEY",booktabs=F) %>%
kable_styling(latex_options = c("striped", "scale_down"))
```

Table 2: Centrality for Word Pair 2: FOOD and MONEY

Betweeness	Page Rank	Eigen Vector
MONEY	MONEY	MONEY
FOOD	FOOD	PAY
BREAD	EAT	COST
WAITRESS	PAY	FOOD
BUY	DINNER	EXPENSE

Ans) For Word Pair 2: FOOD and MONEY, the Betweeness Centrality,Eigen Vector Centrality and Page Rank Centrality gives top 5 values as mentioned in above code.It is observed that the Word 'MONEY','FOOD' are common for all centralities.

Betweeness Centrality for Word Pair 3 RACE and WIN

```
between3<- betweenness(topic_graph2, weights=NA)
n <-5
# the vertex ids of the top n betweenness scores
topn3 <- order(between3, decreasing=TRUE)[1:n]

# the top n vertices
topn3 <-names(V(topic_graph2)[topn3])
```

Page Rank Centrality for Word Pair 3 RACE and WIN

```
pr3<- page_rank(topic_graph2,directed = TRUE, damping = 0.85,weights = NA)$vector

n <-5
# the vertex ids of the top n scores
top_n_pr3 <- order(pr3, decreasing=TRUE)[1:n]

# the top n vertices
top_n_pr3<-names(V(topic_graph2)[top_n_pr3])
```

Eigenvector Centrality for Word Pair 3 RACE and WIN

```
eigen3<- eigen_centrality(topic_graph2, weights=NA)$vector
n <- 5
# the vertex ids of the top n scores
```

Table 3: Centrality for Word Pair 3

Betweeness	Page Rank	Eigen Vector
RACE	WIN	WIN
WIN	RACE	SUCCESS
CAR	WINNER	OVERCOME
WINNER	LOSE	DEFEAT
DRAG	MONEY	CONQUEST

```
top_n_eigen3 <- order(eigen3, decreasing=TRUE)[1:n]

# the top n vertices
top_n_eigen3<-names(V(topic_graph2)[top_n_eigen3])
```

Table for Centralities of Word Pair3

```
wordpair3 <- data.frame(topn3,top_n_pr3,top_n_eigen3)
names(wordpair3) <- c("Betweeness", "Page Rank", "Eigen Vector")
kable(wordpair3,caption = "Centrality for Word Pair 3",booktabs=F) %>%
kable_styling(latex_options  =  c("striped",  "scale_down"))
```

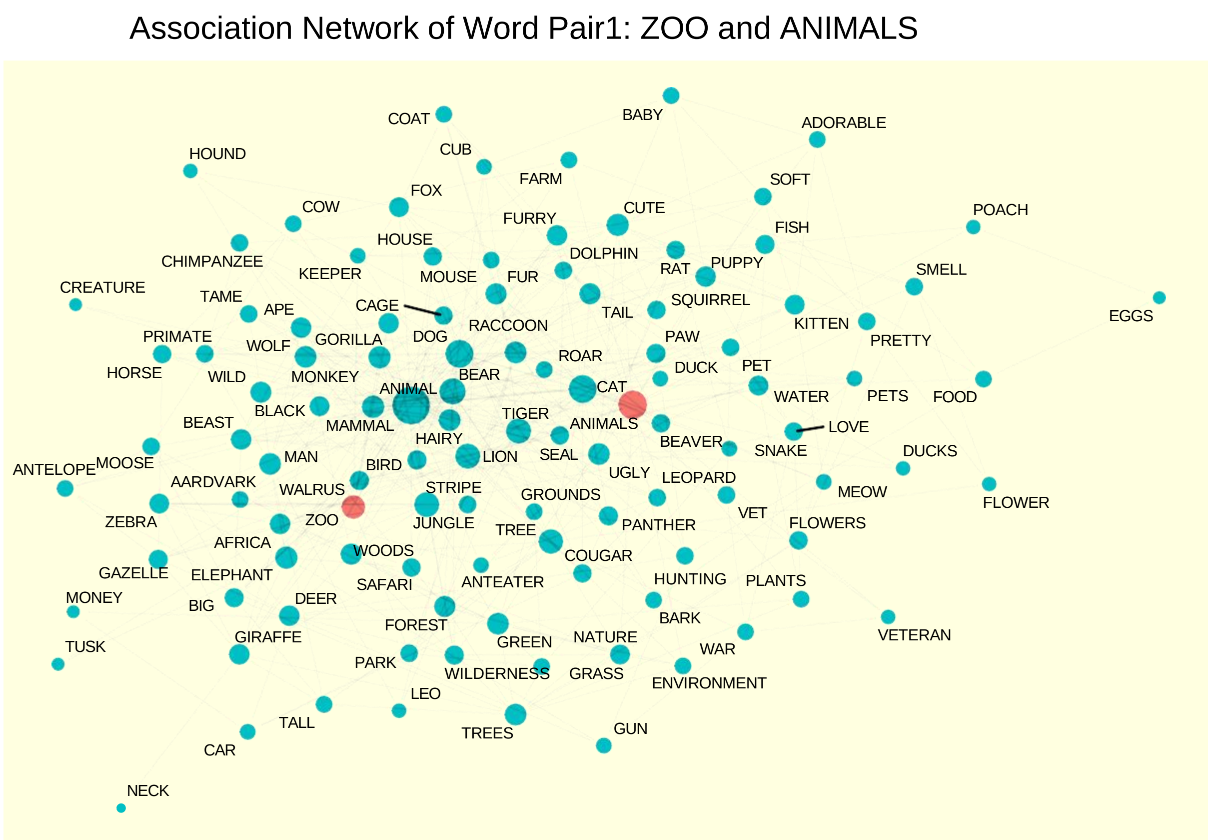
Ans) For Word Pair 3: RACE and WIN, the Betweeness Centrality,Eigen Vector Centrality and Page Rank Centrality gives top 5 values as mentioned in above code.It is observed that the Word 'WINNER' is common for Betweeness and Page Rank centralities.

Q.1.3 Visualize the Association Network

Word Pair 1 - ANIMALS and ZOO

```
set.seed(2323)
vertex_size <- 2 + degree(topic_graph)/10
colour_vertices <- ifelse(names(V(topic_graph)) %in% target_node_names, "#090088", "#FFFF00")

ggraph(topic_graph, layout = "lgl" ) +
geom_node_point(aes(size = vertex_size, color = colour_vertices)) +
geom_node_text(aes(label = name), repel = TRUE, size = 2.2)+
geom_edge_link(start_cap = circle(2, "mm"),
end_cap = circle(1.5, "mm"),
edge_width = 0.3,
```

```
alpha = 0.1) +
ggtitle("Association Network of Word Pair1: ZOO and ANIMALS")+
theme(legend.position = "none",
panel.background = element_rect(fill = "#FFFFE0"),
plot.title = element_text(size=12, hjust = 0.3))
```

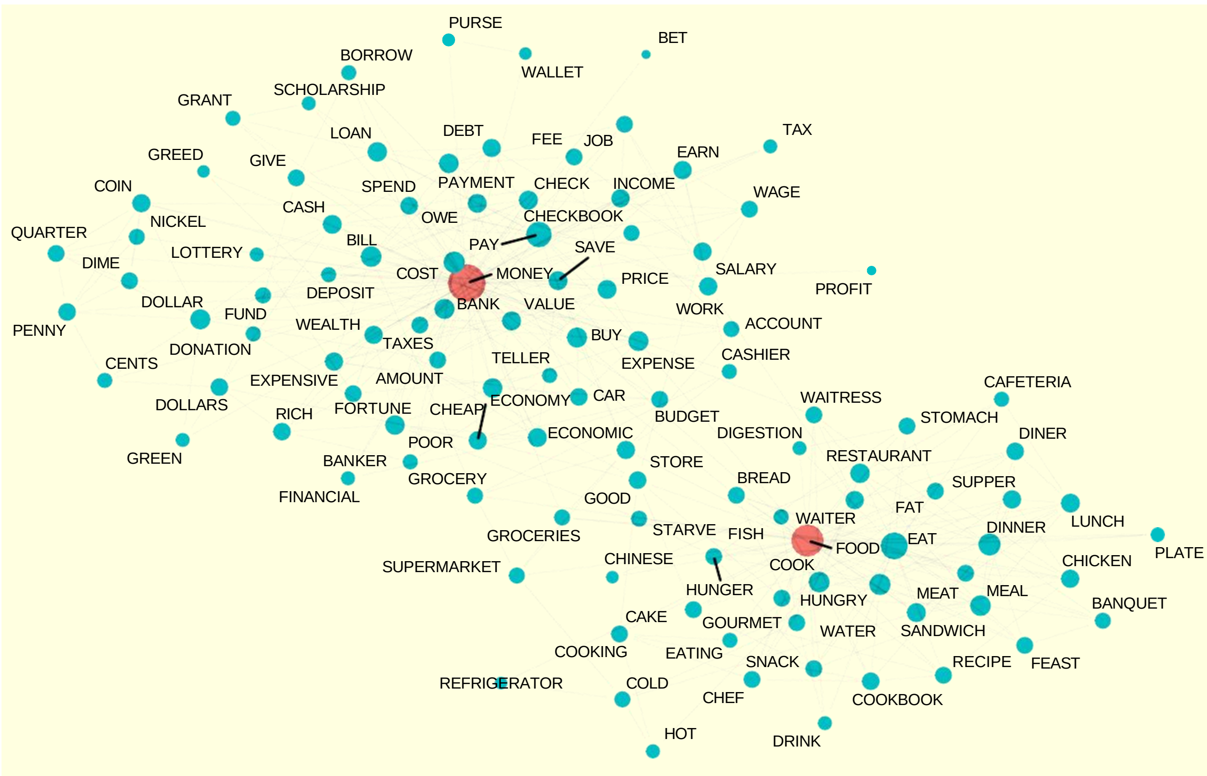


Word Pair 2 - FOOD and MONEY

```
set.seed(2323)

vertex_size <- 2 + degree(topic_graph1)/10

#set colour
colour_vertices <- ifelse(names(V(topic_graph1)) %in% target_node_names2, "#090088", "#FFFF00")

ggraph(topic_graph1, layout = "lgl" ) +
geom_node_point(aes(size = vertex_size, color = colour_vertices)) +
geom_node_text(aes(label = name), repel = TRUE, size = 2.2)+
geom_edge_link(start_cap = circle(2, "mm"),
#set edge at start of circle
end_cap = circle(1.5, "mm"),
edge_width = 0.3,
alpha = 0.1) +
```

```
ggtitle("Association Network of Word Pair2: FOOD and MONEY")+
theme(legend.position = "none",
#yellow background set
panel.background = element_rect(fill = "#FFFFE0"),
plot.title = element_text(size=12, hjust = 0.3))
```

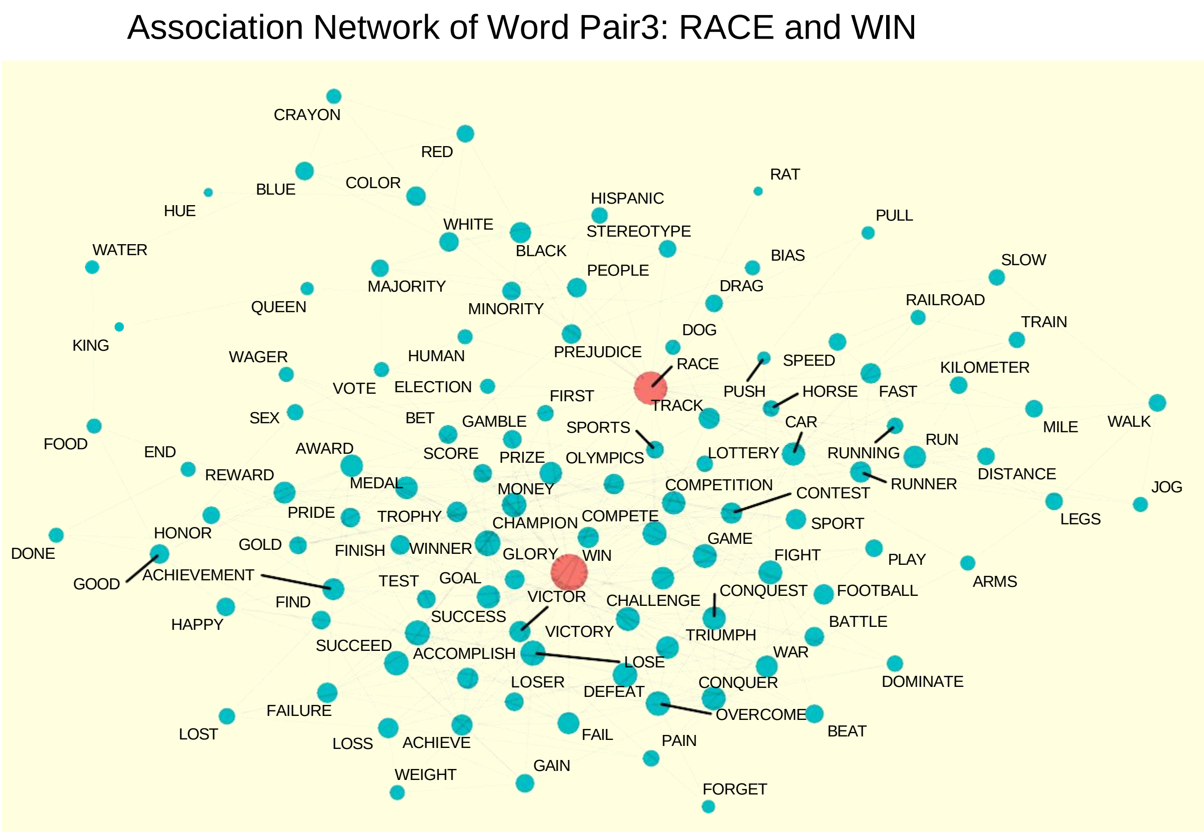
Association Network of Word Pair2: FOOD and MONEY



WORD PAIR 3 - RACE and WIN

```
set.seed(2323)
size_vertex <- 2.0 + degree(topic_graph2)/10

#set colour
colour_vertices <- ifelse(names(V(topic_graph2)) %in% target_node_names3, "#090088", "#FFFF00")

ggraph(topic_graph2, layout = "lgl" ) +
geom_node_point(aes(size = size_vertex, color = colour_vertices)) +
geom_node_text(aes(label = name), repel = TRUE, size = 2.2)+
#set edge at start of circle
geom_edge_link(start_cap = circle(2, "mm"),
end_cap = circle(1.5, "mm"),
edge_width = 0.3,
alpha = 0.1) +
ggtitle("Association Network of Word Pair3: RACE and WIN")+
```

```
theme(legend.position = "none",

#yellow background set
panel.background = element_rect(fill = "#FFFFE0"),
plot.title = element_text(hjust = 0.3))
```



For all 3 Word Pairs a visualization of the association network is presented above with the help of ggraph. The target Word Pairs are coloured in Red and the remaining words generated by personalized page rank are highlighted in blue. The layout considered for the graph is large graph layout(lgl).

Part 2 Q.2.1

Louvain Community Detection Algorithm

PAIR 1 - ANIMALS and ZOO

```
set.seed(3251)
louvain.community1<-cluster_louvain(topic_graph)

# or I could just do
mod_max1<- modularity(louvain.community1)
cat("Modularity Score is:",round(mod_max1,2))
```

```
## Modularity Score is: 0.6
```

```r
# returns the number of communities at the optimal modularity value of
k_opt1<- length(louvain.community1)
cat("\nNumber of Communities formed are:",k_opt1)
```

```
##
## Number of Communities formed are: 12
```

PAIR 1 - ANIMALS and ZOO with weights = NA

```r
set.seed(3251)
louvain.community1<-cluster_louvain(topic_graph,weights = NA)

# or I could just do
mod_max1<- modularity(louvain.community1)
cat("Modularity Score is:",round(mod_max1,2))
```

```
## Modularity Score is: 0.37
```

```r
# returns the number of communities at the optimal modularity value of
k_opt1<- length(louvain.community1)
cat("\nNumber of Communities formed are:",k_opt1)
```

```
##
## Number of Communities formed are: 6
```

PAIR 2 - FOOD and MONEY

```r
set.seed(3251)
louvain.community2<-cluster_louvain(topic_graph1)

# or I could just do
mod_max2<- modularity(louvain.community2)
cat("Modularity Score is:",round(mod_max2,2))
```

```
## Modularity Score is: 0.54
```

```r
# returns the number of communities at the optimal modularity value
k_opt2<- length(louvain.community2)
cat("\nNumber of Communities formed are:",k_opt2)
```

```
##
## Number of Communities formed are: 6
```

PAIR 3 - RACE and WIN

```r
set.seed(3251)
louvain.community3<-cluster_louvain(topic_graph2)

# in contrast to previous approach I use the communities object to get the max modularity

# or I could just do
mod_max3<-   modularity(louvain.community3)
cat("Modularity Score is:",round(mod_max3,2))
```

```
## Modularity Score is: 0.7
```

```r
# returns the number of communities at the optimal modularity value
k_opt3<- length(louvain.community3)
cat("\nNumber of Communities formed are:",k_opt3)
```

```
##
## Number of Communities formed are: 12
```

Fast Greedy Community Detection Algorithm

PAIR 1 - ANIMALS and ZOO

```r
set.seed(4292)

fast_greedy1<- cluster_fast_greedy(topic_graph)

# the max modularity found by the algorithm
mod_max_fg1<- modularity(fast_greedy1)
cat("Modularity Score is:",round(mod_max_fg1,2))
```

```
## Modularity Score is: 0.6
```

```r
# returns the number of communities at the optimal modularity value
k_opt_fg1<- length(fast_greedy1)
cat("\nNumber of Communities formed are:",k_opt_fg1)
```

```
##
## Number of Communities formed are: 11
```

PAIR 2 - FOOD and MONEY

```r
set.seed(4292)

fast_greedy2<- cluster_fast_greedy(topic_graph1)

# the max modularity found by the algorithm
mod_max_fg2<- modularity(fast_greedy2)
cat("Modularity Score is:",round(mod_max_fg2,2))
```

```
## Modularity Score is: 0.54
```

```r
# returns the number of communities at the optimal modularity value of
k_opt_fg2<- length(fast_greedy2)
cat("\nNumber of Communities formed are:",k_opt_fg2)
```

```
##
## Number of Communities formed are: 7
```

PAIR 3 - RACE and WIN

```r
set.seed(4292)

fast_greedy3<- cluster_fast_greedy(topic_graph2)

# the max modularity found by the algorithm
mod_max_fg3<- modularity(fast_greedy3)
cat("Modularity Score is:",round(mod_max_fg3,2))
```

```
## Modularity Score is: 0.7
```

```r
# returns the number of communities at the optimal modularity value of
k_opt_fg3<- length(fast_greedy3)
cat("\nNumber of Communities formed are:",k_opt_fg3)
```

```
##
## Number of Communities formed are: 13
```

WalkTrap Community Detection Algorithm

PAIR 1 - ANIMALS and ZOO

```r
set.seed(3231)

walk_trap1<- cluster_walktrap(topic_graph)

# the max modularity
mod_max_wt1<-modularity(walk_trap1)
cat("Modularity Score is:",round(mod_max_wt1,2))
```

```
## Modularity Score is: 0.58
```

```r
# returns the number of communities at the optimal modularity value
k_wt1<- length(walk_trap1)
cat("\nNumber of Communities formed are:",k_wt1)
```

```
##
## Number of Communities formed are: 12
```

```r
# membership vector
membership<-  membership(walk_trap1)
```

PAIR 2 - FOOD and MONEY

```r
set.seed(3231)

walk_trap2<- cluster_walktrap(topic_graph1)

# the max modularity
mod_max_wt2<-modularity(walk_trap2)
cat("Modularity Score is:",round(mod_max_wt2,2))
```

```
## Modularity Score is: 0.53
```

```r
# returns the number of communities at the optimal modularity value
k_wt2<- length(walk_trap2)
cat("\nNumber of Communities formed are:",k_wt2)
```

```
##
## Number of Communities formed are: 9
```

```r
# membership vector
membership<- membership(walk_trap2)
```

PAIR 3 - RACE and WIN

```r
set.seed(3231)

walk_trap3<- cluster_walktrap(topic_graph2)

# the max modularity
mod_max_wt3<-modularity(walk_trap3)
cat("Modularity Score is:",round(mod_max_wt3,2))
```

```
## Modularity Score is: 0.68
```

```r
# returns the number of communities at the optimal modularity value
k_wt3<- length(walk_trap3)
cat("\nNumber of Communities formed are:",k_wt3)
```

```
##
## Number of Communities formed are: 17
```

```r
# membership vector
membership<- membership(walk_trap3)
```

Table 4: Cluster and Size Table for Word Pair 1 ANIMALS and ZOO

Sr.No	Size	Cluster
1	26	AARDVARK, ANIMAL , ANTEATER, ZOO , WILD , FARM , SAFARI , AFRICA , JUNGLE , ELEPHANT, LION , COW , HORSE , CAGE , TIGER , PANTHER , GAZELLE , LEO , NECK , CUB , COUGAR , LEOPARD , GIRAFFE , ROAR , ZEBRA , TAME
2	21	[illegible] ANIMALS , SOFT , FURRY , RACCOON , RAT , BEAVER , TAIL , SNAKE , COAT , MOUSE , FOX , DUCKS
3	9	BLACK , MAN , BEAST , MONKEY , APE , GORILLA , HAIRY , CHIMPANZEE, PRIMATE
4	24	[illegible] ANTELOPE , DEER , MOOSE , FLOWERS , BARK , MEOW , WOODS , FOREST , GROUNDS , WILDERNESS , PLANTS , KEEPER
5	12	FOOD , WATER , MAMMAL , SMELL , EGGS , FISH , FLOWER , SEAL , DOLPHIN, TUSK , POACH , WALRUS
6	18	HOUSE , LOVE , GUN , BIRD , CAT , WAR , DOG , PET , PETS , PAW , HOUND , DUCK , WOLF , CREATURE, HUNTING , STRIPE , VETERAN , VET

Ans) In my opinion out of all the community detection algorithms tested, the algorithm that works best for Pair 1,Pair 2 and Pair 3 is the Louvain Community Detection algorithm as this algorithm provides maximum modularity(Word Pair 1- 0.6, Word Pair 2- 0.54 ,Word Pair 3- 0.71). Modularity is a measure of how likely a subgraph can become a community.It is measure of how the subgraph differs from an equivalent random subgraph in terms of intra-community links.

The Louvain Alogirthm uses an ideal approach of modularity maximisation on small communities and then merges them to form larger communities. Also, Louvain algorithm is a multi-level optimization algorithm which performs better in terms of computation time as compared to other algorithms.Louvain algorithm stops processing when a local maxima of modularity is achieved.

Walktrap Clusturing Algorithm gives less modularity as compared to Lovain Community Detection Algorithm (Word Pair 1- 0.58, Word Pair 2- 0.53 ,Word Pair 3- 0.68) and Fast Greedy even though it gives same modularity as the louvain community detection algorithm is not an ideal approach as it is known to suffer from the resolution limit. Communities below a particular size are merged with other communities.

Community detection algorithm when used with edge weights of the Word Pairs subgraph provides higher modularity as compared to parameter set to weights = NA (example given above)and thus for the below codes edge weights of subgraphs are considered.

Q.2.2 Tables listing each Community

Word Pair 1 ANIMALS and ZOO

```r
length_community1 <- length(louvain.community1)

df_cluster1 <- as.data.frame(louvain.community1[1:length_community1])
size_cluster1 <- sizes(louvain.community1)

combine_df_cluster1 <- cbind(size_cluster1,df_cluster1)

kable(combine_df_cluster1,col.names = c("Sr.No","Size","Cluster"),
      caption = "Cluster and Size Table for Word Pair 1 ANIMALS and ZOO",booktabs=F) %>%
kable_classic_2(html_font = "Times New Roman",latex_options = c("striped","scale_down"))
```

Word Pair 2 FOOD and MONEY

```r
length_community2 <- length(louvain.community2)

df_cluster2 <- as.data.frame(louvain.community2[1:length_community2])
```

Table 5: Cluster and Size Table for Word Pair 2 FOOD and MONEY

Sr.No	Size	Cluster
1	40	STOMACH , GOOD , FOOD , STORE , LUNCH , CAKE , FAT , BANQUET , EAT , REFRIGERATOR, COOK , COOKING , CHEF , PLATE , MEAT , GROCERIES , BREAD , CHICKEN , DINNER , FEAST , MEAL , WAITER , WAITRESS , HUNGRY , SANDWICH , RESTAURANT , DINER , CAFETERIA , SUPERMARKET , GROCERY , CHINESE , SNACK , COOKBOOK , RECIPE , HUNGER , EATING , DIGESTION , SUPPER , STARVE , GOURMET
3	47	MONEY , GIVE , ACCOUNT , BANK , CHECKBOOK , SAVE , EARN , BUY , GREEN , PAY , FEE , PRICE , CASH , BET , FINANCIAL , GREED , AMOUNT , COST , VALUE , CHECK , GRANT , PURSE , TELLER , LOAN , BANKER , BORROW , PROFIT , OWE , WALLET , BUDGET , ECONOMY , SPEND , DEPOSIT , FUND , CASHIER , LOTTERY , DONATION , DEBT , TAXES , TAX , EXPENSE , SALARY , ECONOMIC , FORTUNE , SCHOLARSHIP, INCOME , WAGE
3	[illegible]	CAR , DOLLAR , BILL , PAYMENT, PENNY , DIME , COIN , CENTS , DOLLARS, NICKEL , QUARTER

Table 6: Cluster and Size Table for Word Pair 3 RACE and WIN

Sr.No	Size	Cluster
1	18	PAIN , SEX , GOOD , FOOD , ACCOMPLISH , FINISH , ACHIEVE , GOAL , SUCCEED , DONE , ACHIEVEMENT, HAPPY , SUCCESS , FAIL , FAILURE , TEST , END , SCORE
2	10	SPORT , OLYMPICS , PLAY , GAME , SPORTS , COMPETITION, FOOTBALL , CONTEST , CHALLENGE , COMPETE
3	16	LOST , BEAT , WIN , GAIN , CONQUER , WEIGHT , DEFEAT , FORGET , FIND , VICTORY , LOSE , OVERCOME, CONQUEST, TRIUMPH , DOMINATE, LOSS
4	23	BLACK , PEOPLE , WATER , RACE , BLUE , RED , HUMAN , WHITE , DOG , HORSE , COLOR , RAT , BIAS , PREJUDICE , STEREOTYPE, PUSH , PULL , CRAYON , HUE , MAJORITY , DRAG , HISPANIC , MINORITY
5	14	MONEY , AWARD , HONOR , PRIDE , BET , TROPHY , PRIZE , REWARD , MEDAL , GOLD , GAMBLE , WAGER , LOTTERY, GLORY
6	3	FIGHT , WAR , BATTLE
7	5	SPEED , FAST , CAR , SLOW , RUNNING
8	2	KING , QUEEN
9	5	WINNER , FIRST , LOSER , CHAMPION, VICTOR
10	10	MILE , RUN , RUNNER , TRAIN , WALK , TRACK , RAILROAD , DISTANCE , JOG , KILOMETER
11	2	ARMS, LEGS
12	2	VOTE , ELECTION

```
size_cluster2 <- sizes(louvain.community2)

combine_df_cluster2 <- cbind(size_cluster2,df_cluster2)

kable(combine_df_cluster2,col.names    =    c("Sr.No","Size","Cluster"),
      caption = "Cluster and Size Table for Word Pair 2 FOOD and MONEY",booktabs=F) %>%
kable_classic_2(html_font = "Times New Roman",latex_options = c("striped","scale_down"))
```

Word Pair 3 RACE and WIN

```
length_community3 <- length(louvain.community3)

df_cluster3 <- as.data.frame(louvain.community3[1:length_community3])
size_cluster3 <- sizes(louvain.community3)

combine_df_cluster3 <- cbind(size_cluster3,df_cluster3)

kable(combine_df_cluster3,col.names = c("Sr.No","Size","Cluster"),
      caption = "Cluster and Size Table for Word Pair 3 RACE and WIN",booktabs=F) %>%
kable_classic_2(html_font = "Times New Roman",latex_options = c("striped","scale_down"))
```

Q.2.3 1 Word Labels for Communities

Word Pair 1: ANIMALS and ZOO

```
final_labels1 <- c()
for(i in 1:length_community1)
{
  subgraphs1<- induced.subgraph(topic_graph,louvain.community1[[i]])
  subgraph1_degree <- degree(subgraphs1)
  sort_degree1 <- sort(subgraph1_degree,decreasing = T)[1]
  labels1 <- names(sort_degree1)
  final_labels1 <- c(final_labels1,labels1)
```

Table 7: Table for Word Pair 1 ANIMALS and ZOO Labels

Sr.No	Size	Cluster	Labels
1	26	AARDVARK, ANIMAL , ANTEATER, ZOO , WILD , FARM , SAFARI , AFRICA , JUNGLE , ELEPHANT, LION , COW , HORSE , CAGE , TIGER , PANTHER , GAZELLE , LEO , NECK , CUB , COUGAR , LEOPARD , GIRAFFE , ROAR , ZEBRA , TAME	ANIMAL
2	21	UGLY , FUR , SQUIRREL, PRETTY , CUTE , ADORABLE, BABY , PUPPY , KITTEN , ANIMALS , SOFT , FURRY , RACCOON , RAT , BEAVER , TAIL , SNAKE , COAT , MOUSE , FOX , DUCKS	CUTE
3	9	BLACK , MAN , BEAST , MONKEY , APE , GORILLA , HAIRY , CHIMPANZEE, PRIMATE	APE
4	24	MONEY , TREES , CAR , TREE , GREEN , BIG , GRASS , ENVIRONMENT, TALL , PARK , BEAR , NATURE , ANTELOPE , DEER , MOOSE , FLOWERS , BARK , MEOW , WOODS , FOREST , GROUNDS , WILDERNESS , PLANTS , KEEPER	TREES
5	12	FOOD , WATER , MAMMAL , SMELL , EGGS , FISH , FLOWER , SEAL , DOLPHIN, TUSK , POACH , WALRUS	FISH
6	18	HOUSE , LOVE , GUN , BIRD , CAT , WAR , DOG , PET , PETS , PAW , HOUND , DUCK , WOLF , CREATURE, HUNTING , STRIPE , VETERAN , VET	DOG

Table 8: Table for Word Pair 2 FOOD and MONEY Labels

Sr.No	Size	Cluster	Labels
1	40	STOMACH , GOOD , FOOD , STORE , LUNCH , CAKE , FAT , BANQUET , EAT , REFRIGERATOR, COOK , COOKING , CHEF , PLATE , MEAT , GROCERIES , BREAD , CHICKEN , DINNER , FEAST , MEAL , WAITER , WAITRESS , HUNGRY , SANDWICH , RESTAURANT , DINER , CAFETERIA , SUPERMARKET , GROCERY , CHINESE , SNACK , COOKBOOK , RECIPE , HUNGER , EATING , DIGESTION , SUPPER , STARVE , GOURMET	FOOD
3	47	MONEY , GIVE , ACCOUNT , BANK , CHECKBOOK , SAVE , EARN , BUY , GREEN , PAY , FEE , PRICE , CASH , BET , FINANCIAL , GREED , AMOUNT , COST , VALUE , CHECK , GRANT , PURSE , TELLER , LOAN , BANKER , BORROW , PROFIT , OWE , WALLET , BUDGET , ECONOMY , SPEND , DEPOSIT , FUND , CASHIER , LOTTERY , DONATION , DEBT , TAXES , TAX , EXPENSE , SALARY , ECONOMIC , FORTUNE , SCHOLARSHIP, INCOME , WAGE	MONEY
5	11	CAR , DOLLAR , BILL , PAYMENT, PENNY , DIME , COIN , CENTS , DOLLARS, NICKEL , QUARTER	DOLLAR

```
}
final_labels1  <-  as.data.frame(final_labels1)
label_df_cluster1 <- cbind(combine_df_cluster1,final_labels1)

 knitr::kable(label_df_cluster1,col.names    =    c("Sr.No","Size","Cluster","Labels"),
         caption  =  "Table  for  Word  Pair  1  ANIMALS  and  ZOO  Labels")  %>%
 kable_classic_2(html_font  =  "Times  New  Roman",latex_options  =  c("striped","scale_down"))
```

Word Pair 2: FOOD AND MONEY

```
final_labels2 <- c()
for(i  in  1:length_community2)
{
   subgraph2<- induced.subgraph(topic_graph1,louvain.community2[[i]])
   subgraph2_degree <- degree(subgraph2)
   sort_degree2 <- sort(subgraph2_degree,decreasing = T)[1]
   labels2  <-  names(sort_degree2)
   final_labels2  <-  c(final_labels2,labels2)
}

label_df_cluster2  <-  cbind(combine_df_cluster2,final_labels2)


 knitr::kable(label_df_cluster2,col.names    =    c("Sr.No","Size","Cluster","Labels"),
         caption  =  "Table  for  Word  Pair  2  FOOD  and  MONEY  Labels")  %>%
 kable_classic_2(html_font  =  "Times  New  Roman",latex_options  =  c("striped","scale_down"))
```

Word Pair 3: RACE and WIN

```
final_labels3 <- c()
for(i  in  1:length_community3)
{
   subgraph3<- induced.subgraph(topic_graph2,louvain.community3[[i]])
   subgraph3_degree <- degree(subgraph3)
   sort_degree3 <- sort(subgraph3_degree,decreasing = T)[1]
   labels3  <-  names(sort_degree3)
   final_labels3  <-  c(final_labels3,labels3)
```

Table 9: Table for Word Pair 3 RACE and WIN Labels

Sr.No	Size	Cluster	Labels
1	18	PAIN , SEX , GOOD , FOOD , ACCOMPLISH , FINISH , ACHIEVE , GOAL , SUCCEED , DONE , ACHIEVEMENT, HAPPY , SUCCESS , FAIL , FAILURE , TEST , END , SCORE	SUCCEED
2	10	SPORT , OLYMPICS , PLAY , GAME , SPORTS , COMPETITION, FOOTBALL , CONTEST , CHALLENGE , COMPETE	GAME
3	16	LOST , BEAT , WIN , GAIN , CONQUER , WEIGHT , DEFEAT , FORGET , FIND , VICTORY , LOSE , OVERCOME, CONQUEST, TRIUMPH , DOMINATE, LOSS	WIN
4	23	BLACK , PEOPLE , WATER , RACE , BLUE , RED , HUMAN , WHITE , DOG , HORSE , COLOR , RAT , BIAS , PREJUDICE , STEREOTYPE, PUSH , PULL , CRAYON , HUE , MAJORITY , DRAG , HISPANIC , MINORITY	RACE
5	14	MONEY , AWARD , HONOR , PRIDE , BET , TROPHY , PRIZE , REWARD , MEDAL , GOLD , GAMBLE , WAGER , LOTTERY, GLORY	MONEY
6	3	FIGHT , WAR , BATTLE	FIGHT
7	5	SPEED , FAST , CAR , SLOW , RUNNING	FAST
8	2	KING , QUEEN	KING
9	5	WINNER , FIRST , LOSER , CHAMPION, VICTOR	WINNER
10	10	MILE , RUN , RUNNER , TRAIN , WALK , TRACK , RAILROAD , DISTANCE , JOG , KILOMETER	RUN
11	2	ARMS, LEGS	ARMS
12	2	VOTE , ELECTION	VOTE

Table 10: Table for Word Pair 1 ANIMALS and ZOO Interpretations

Sr.No	Size	Cluster	Labels	Interpretation
1	26	AARDVARK, ANIMAL , ANTEATER, ZOO , WILD , FARM , SAFARI , AFRICA , JUNGLE , ELEPHANT, LION , COW , HORSE , CAGE , TIGER , PANTHER , GAZELLE , LEO , NECK , CUB , COUGAR , LEOPARD , GIRAFFE , ROAR , ZEBRA , TAME	ANIMAL	Kinds of Animals in Zoo
[illegible]	9	BLACK , MAN , BEAST , MONKEY , APE , GORILLA , HAIRY , CHIMPANZEE, PRIMATE	APE	Fur of Animal can be used to make products like COAT
[illegible]	12	FOOD , WATER , MAMMAL , SMELL , EGGS , FISH , FLOWER , SEAL , DOLPHIN, TUSK , POACH , WALRUS	FISH	Evolution of Animals from Primates to Man
[illegible]	26	AARDVARK, ANIMAL , ANTEATER, ZOO , WILD , FARM , SAFARI , AFRICA , JUNGLE , ELEPHANT, LION , COW , HORSE , CAGE , TIGER , PANTHER , GAZELLE , LEO , NECK , CUB , COUGAR , LEOPARD , GIRAFFE , ROAR , ZEBRA , TAME	ANIMAL	Animals present in Water
[illegible]	9	BLACK , MAN , BEAST , MONKEY , APE , GORILLA , HAIRY , CHIMPANZEE, PRIMATE	APE	Parts and Sound of Pet Animal Dog
[illegible]	12	FOOD , WATER , MAMMAL , SMELL , EGGS , FISH , FLOWER , SEAL , DOLPHIN, TUSK , POACH , WALRUS	FISH	Carnivorous Animals of Zoo

```
}

label_df_cluster3 <- cbind(combine_df_cluster3,final_labels3)

 knitr::kable(label_df_cluster3,col.names  =  c("Sr.No","Size","Cluster","Labels"),
        caption = "Table for Word Pair 3 RACE and WIN Labels") %>%
 kable_classic_2(html_font  =  "Times  New  Roman",latex_options  =  c("striped","scale_down"))
```

Q.2.4 Brief Interpretations

```
 wordp1_interpret <- c('Kinds of Animals in Zoo',
                     'Physical Appearance of Animals',
                     'Fur of Animal can be used to make products like COAT',
                      'Types  of  Animals  in  Zoo-  WILD/TAME',
                      'Evolution of Animals from Primates to Man',
                      'Present  in  Zoo-  Grass,Tree,Flower etc',
                      'Animals  present  in  Water',
                      'Animals in the Cat family,It also provides the cat sound and
                       what it eats-Mouse',
                      'Parts and Sound of Pet Animal Dog',
                      'Big  and  Tall  Animals  of  Zoo',
                      'Carnivorous  Animals  of  Zoo',
                      'Irrelevant  to  the  Word  Pair  but  both  words  are  related  to  eachother-  Veterans  go


interpret_df_cluster1 <- cbind(label_df_cluster1,wordp1_interpret)

 knitr::kable(interpret_df_cluster1,col.names  =  c("Sr.No","Size","Cluster",
                                                    "Labels",  "Interpretation"),
        caption = "Table for Word Pair 1 ANIMALS and ZOO Interpretations",
        )  %>%
 kable_classic_2(html_font  =  "Times  New  Roman",latex_options  =  c("striped","scale_down"))
```

Table 11: Table for Word Pair 2 FOOD and MONEY Interpretations

Sr.No	Size	Cluster	Labels	Interpretation
1	40	STOMACH , GOOD , FOOD , STORE , LUNCH , CAKE , FAT , BANQUET , EAT , REFRIGERATOR, COOK , COOKING , CHEF , PLATE , MEAT , GROCERIES , BREAD , CHICKEN , DINNER , FEAST , MEAL , WAITER , WAITRESS , HUNGRY , SANDWICH , RESTAURANT , DINER , CAFETERIA , SUPERMARKET , GROCERY , CHINESE , SNACK , COOKBOOK , RECIPE , HUNGER , EATING , DIGESTION , SUPPER , STARVE , GOURMET	FOOD	Food Cuisines,Meals of the Day,Places where and by whom food is served
2	2	WORK, JOB	WORK	Working/Doing Job for earning Money
3	47	MONEY , GIVE , ACCOUNT , BANK , CHECKBOOK , SAVE , EARN , BUY , GREEN , PAY , FEE , PRICE , CASH , BET , FINANCIAL , GREED , AMOUNT , COST , VALUE , CHECK , GRANT , PURSE , TELLER , LOAN , BANKER , BORROW , PROFIT , OWE , WALLET , BUDGET , ECONOMY , SPEND , DEPOSIT , FUND , CASHIER , LOTTERY , DONATION , DEBT , TAXES , TAX , EXPENSE , SALARY , ECONOMIC , FORTUNE , SCHOLARSHIP, INCOME , WAGE	MONEY	Banking Terminologies related to Money
4	5	WATER, HOT , DRINK, COLD , FISH	WATER	Drinking Hot or Cold beverages,also fish being a water animal and food is thus present in the community
5	11	CAR , DOLLAR , BILL , PAYMENT, PENNY , DIME , COIN , CENTS , DOLLARS, NICKEL , QUARTER	DOLLAR	Bill Payments and Different Currencies
6	5	WEALTH , POOR , RICH , CHEAP , EXPENSIVE	POOR	Objectifying people or objects with Money

```
  wordp2_interpret <- c('Food Cuisines,Meals of the Day,Places where and by whom
                          food is served',
                        'Working/Doing Job for earning Money',
                        'Banking Terminologies related to Money',
                        'Drinking Hot or Cold beverages,also fish being a water
                        animal and food is thus present in the community ',
                        'Bill Payments and Different Currencies',
                        'Objectifying people or objects with Money')

interpret_df_cluster2 <- cbind(label_df_cluster2,wordp2_interpret)

 knitr::kable(interpret_df_cluster2,col.names = c("Sr.No","Size","Cluster","Labels",
                                                  "Interpretation"),
        caption = "Table for Word Pair 2 FOOD and MONEY Interpretations",
        ) %>%
 kable_classic_2(html_font = "Times New Roman",latex_options = c("striped","scale_down"))
```

Table for Word Pair 3: RACE and WIN

```
   wordp3_interpret <- c('Positive terms for achievements,negative terms for failure,
                           emotions expressing achieving a goal/failure',
                        'Various Sports or Contests',
                        'Most of the words state Win or Loose a Conquest and some
                         state Loose Weight',
                        'Determining someone by Race,Colour,Bias etc. Also due to presence
                         of colours Crayon is present',
                        'Winning a Reward',
                        'Winning a war/battle',
                        'Speed of Car or person running',
                        'Winning a game of Chess,thus mention of a King and Queen/Drag Queen',
                        'Various Terms for Winning, one states Loose',
                        'Different Ways in which a person is covering Distance',
                        'Body Parts required for a Race',
                        'Winning an Election'
)


interpret_df_cluster3 <- cbind(label_df_cluster3,wordp3_interpret)

 knitr::kable(interpret_df_cluster3,col.names = c("Sr.No","Size","Cluster","Labels",
                                                  "Interpretation"),
        caption = "Table for Word Pair 3 RACE and WIN Interpretations",
        ) %>%
 kable_classic_2(html_font = "Times New Roman",latex_options = c("striped", "scale_down"))
```

Table 12: Table for Word Pair 3 RACE and WIN Interpretations

Sr.No	Size	Cluster	Labels	Interpretation
1	18	PAIN , SEX , GOOD , FOOD , ACCOMPLISH , FINISH , ACHIEVE , GOAL , SUCCEED , DONE , ACHIEVEMENT, HAPPY , SUCCESS , FAIL , FAILURE , TEST , END , SCORE	SUCCEED	Positive terms for achievements,negative terms for failure, emotions expressing achieving a goal/failure
2	10	SPORT , OLYMPICS , PLAY , GAME , SPORTS , COMPETITION, FOOTBALL , CONTEST , CHALLENGE , COMPETE	GAME	Various Sports or Contests
3	16	LOST , BEAT , WIN , GAIN , CONQUER , WEIGHT , DEFEAT , FORGET , FIND , VICTORY , LOSE , OVERCOME, CONQUEST, TRIUMPH , DOMINATE, LOSS	WIN	Most of the words state Win or Loose a Conquest and some state Loose Weight
4	23	BLACK , PEOPLE , WATER , RACE , BLUE , RED , HUMAN , WHITE , DOG , HORSE , COLOR , RAT , BIAS , PREJUDICE , STEREOTYPE, PUSH , PULL , CRAYON , HUE , MAJORITY , DRAG , HISPANIC , MINORITY	RACE	Determining someone by Race,Colour,Bias etc. Also due to presence of colours Crayon is present
5	14	MONEY , AWARD , HONOR , PRIDE , BET , TROPHY , PRIZE , REWARD , MEDAL , GOLD , GAMBLE , WAGER , LOTTERY, GLORY	MONEY	Winning a Reward
[illegible]	[illegible]	[illegible]	[illegible]	Winning a war/battle
7	5	SPEED , FAST , CAR , SLOW , RUNNING	FAST	Speed of Car or person running
[illegible]	[illegible]	[illegible]	[illegible]	Winning a game of Chess,thus mention of a King and Queen/Drag Queen
9	5	WINNER , FIRST , LOSER , CHAMPION, VICTOR	WINNER	Various Terms for Winning, one states Loose
[illegible]	[illegible]	[illegible] , RUNNER , TRAIN , WALK , TRACK , RAILROAD , DISTANCE , JOG , KILOMETER	[illegible]	Different ways in which a person is covering Distance
11	2	ARMS, LEGS	ARMS	Body Parts required for a Race
[illegible]	[illegible]	VOTE , ELECTION	VOTE	Winning an Election

Community Clusters are formed with the help of Louvain Clustering Algorithm and presented in the form of a table with the help of kableExtra library. All the words formed even though are of large size are homogenous in nature and if irrelevant are commented in the interpretation section. A word label is generated for each community of every word pair which likely provides the most relevance and gives a clear meaning of the community in a single word. All possible interpretations of those communities are listed in the table above as well.

References:

1) Week 10 Lecture Notes of Web and Network Science

2) Week 3 Lecture Notes of Web and Network Science

3) File CT5113_PersonalisedPR.html from Week 10 of Web and Network Science

4) File Communities_Intro.html from Week 9 of Web and Network Science