WNS Assignment 5

PART 1

Reading the graph

Reference to Code Appendix- read_graph

The first step is to read the graph using the igraph method and make it undirected. I also removed the edges that pointed to the same vertices they emerged from by using simplify() method.

Finding out cue words from the graph

Reference to Code Appendix-scan

In the next step, I read the file- cue.txt using the scan() method. After reading it in a variable I set it as a vertex attribute. This will help us to eliminate those words that are not cue words.After this I deleted the non-cue words from the graph.

Pre-processing I

Reference to Code Appendix- deleteing_vertices

This involved deletion of those nodes that had degree 0. I also scaled the edge weights from 0 to 1 at this point.

edgedeletion() function

Reference to Code Appendix-edgedeletion

I created a function that takes a graph and a numeric value (threshold) as parameter and deletes those edges from the graph that have edge weight less than the threshold. This method is called for each of my target words (ART, KNOWLEDGE, MIND) to delete edges that have weights less than threshold.

Random Walk

Reference to Code Appendix-random_walk

I now create 3 vectors that will store the results of random walks for my target words. Then using a for loop that runs 1000 times, I called the random_walk() method from each target word and got the neighborhood around each target word in these vectors.

vertexdeletion() function

Reference to Code Appendix-vertexdeletion

This method I created takes a graph and the random_walk_vector as input and deletes those vertices that are not in the neighbourhood, by deleting vertices that are not in the random_walk_vector. After calling this method for each of my target words I get 3 graphs. Each of these graphs are a subset of the original graph and consists of only the neighborhood of my target word.

Pre-processing II

After this I delete from each of the three graphs those vertices that have degree=1.

plotgraph() function

Reference to Code Appendix-plotgraph

This function takes an igraph as an input and returns a ggraph plot. Using this function I have plotted association networks for each of my target words.

Authority measures and authority scores for each target word

Reference to Code Appendix-AUTHORITY_MEASURES

I have selected Page Rank algorithm for authority score calculation of target word- ART, this is done with the help of page_rank() method. Similarly for target word-KNOWLEDGE I have calculated the authority score using authority_score() method. Finally for the third target word- MIND I have used degree centrality for calculating the authority. Below is a snippet of the authorities in each target word.

Target Word: ART

	Authority Score
ART	0.0965221
PICTURE	0.0720184
DRAW	0.0432245

Target Word: KNOWLEDGE

	Authority Score
SMART	1.0000000
DUMB	0.5671321
INTELLIGENT	0.4771101

Target Word: MIND

	Authority Score
THINK	14
MIND	13
BRAIN	12

PART 2:

Using Louvain algorithm for community detection

Reference to Code Appendix-clusterdf

I have used Louvain algorithm for community detection which can be done with the help of cluster_louvain() method. This is extremely easy to use and simple and efficient.

I have created a method called clusterdf() which takes an igraph as a parameter and returns a dataframe that consists of Cluster Number, Members of that cluster (Words In our case), Total count of members in that cluster.

I had called this method for each of my target words and got the Cluster Information as shown below

ART

Cluster Number	Members	Count
1	ABILITY SKILL TALENT	3
2	MAKE DESTROY DESIGN CREATE INVENT DEVELOP	6
3	ABSTRACT ART PAINTING DRAWING SCULPTURE CRAFT CREATIVE ARTS CRAFTS MUSEUM WRITING	11
4	HOUSE ARTIST PAINTER PAINT WALL BRUSH PLASTER CREATIVITY	8
5	CANVAS SHOES TENT CANVASS	4
6	GIRL BEAUTY GOD MATERIAL CREATION GODDESS	6
7	DRAW PENCIL WRITE SKETCH COMPOSE SCRIBBLE	6
8	CAMERA PICTURE PHOTO FILM DIAGRAM PORTRAIT	6

KNOWLEDGE

Cluster Number	Members	Count
1	KNOWLEDGE GOOD NEWSPAPER NEWS WISDOM LOOK INFORMATION EXPERIENCE INSIGHT	9
2	MIND THOUGHT THINK HEAD DECIDE MEMORY NEURON NERVE BRAIN SKULL THINKING MENTAL	12
3	STUPID SMART LOGIC GENIUS BRIGHT INTELLIGENCE INTELLIGENT BRILLIANT EXCEPTIONAL WIT CLEVER EINSTEIN GIFTED INTELLECT	14
4	DICTIONARY BOOK ENCYCLOPEDIA READ BRITANNICA	5
5	WORK SCHOOL LEARN TEACH TEACHER STUDY PROFESSOR EDUCATION LEARNING LESSON EDUCATE	11
6	UNDERSTAND ACKNOWLEDGE KNOW RECOGNITION REALIZE SYMPATHETIC UNDERSTANDING COMPREHEND INTUITION PERCEIVE	10

MIND

Cluster Number	Members	Count
1	THOUGHT IDEA OPINION FACT NOTION	5
2	EASY COMPLEX BASIC HARD SIMPLE EASE	6
3	HEAD ATOM CELL NEURON NERVE BRAIN SKULL THINKING	8
4	MIND PSYCHOLOGY CLASS AWAKE AWARE SCIENCE EXPERIMENT CONSCIOUS SUBJECT CONSCIENCE MATTER FEEBLE	12
5	SMART LOGIC INTELLIGENCE IMAGINATION CREATIVITY INTELLECT	6
6	BODY PHYSICAL ANATOMY PHYSIOLOGY MENTAL	5
7	FORGET MEMORY REMEMBER RECALL REMINISCENCE	5
8	THINK WONDER STUDY ANALYZE DREAM GUESS SUPPOSE CONCENTRATE PONDER IMAGINE	10

Interpretation

Reference to Code Appendix-Interpretation

ART CLUSTERS INTERPRETATION

Cluster No.	Interpretation
1	Art as a kind of Talent
2	Art for Renovation
3	Art for Restoration
4	Result of Art like House/Artist/Painter etc
5	Art Canvas
6	Art as in Divine Creation
7	Book Art like Sketching/Scribbling
8	Digital Art

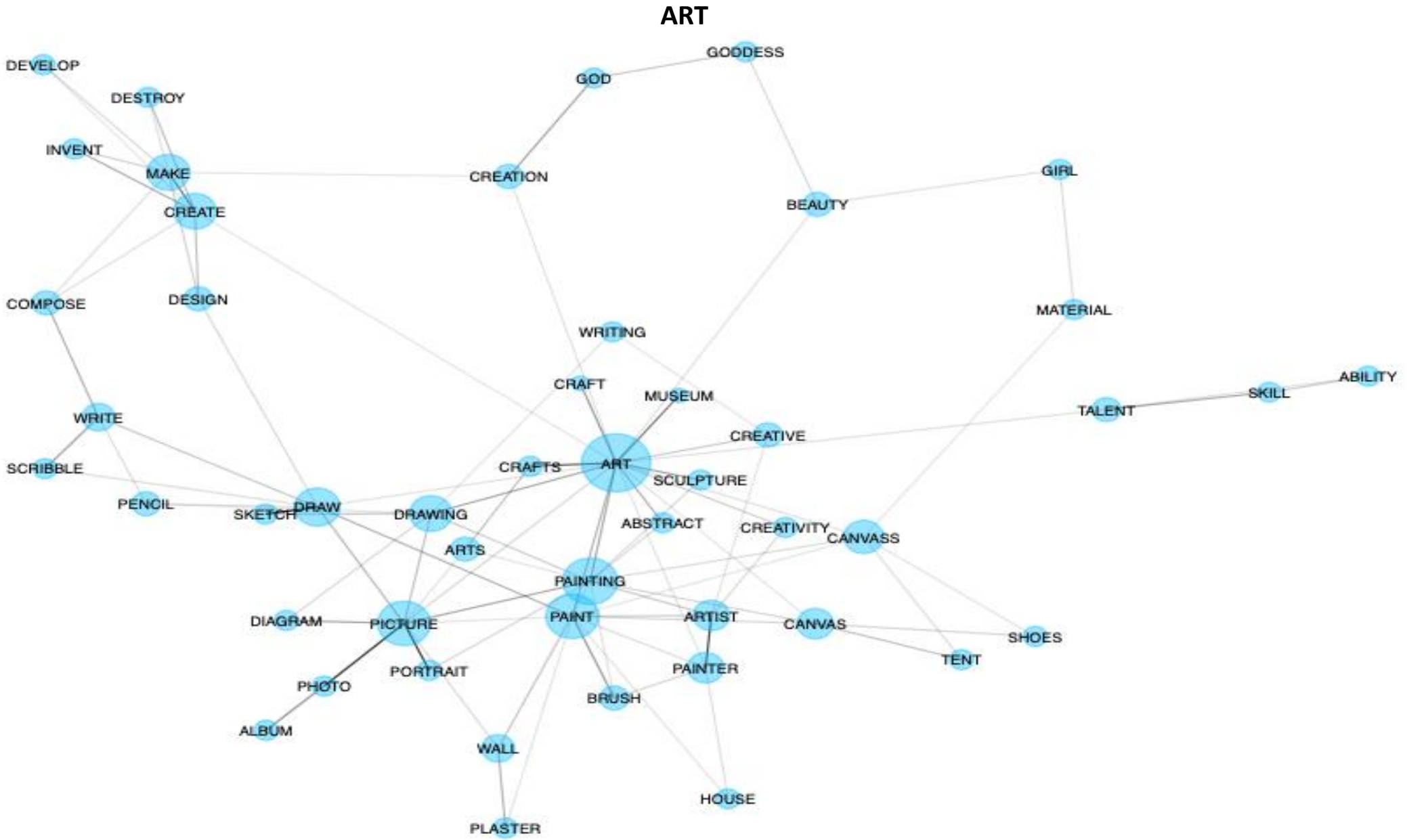
KNOWLEDGE CLUSTERS INTERPRETATION

Cluster No.	Interpretation
1	Getting Knowledge with Experience and Information
2	Where knowledge is probably stored
3	Classification of a person based on the knowedge they have
4	Resources to obtain knowledge
5	Academic and Educational Knowledge
6	Traits of a person having knowldege

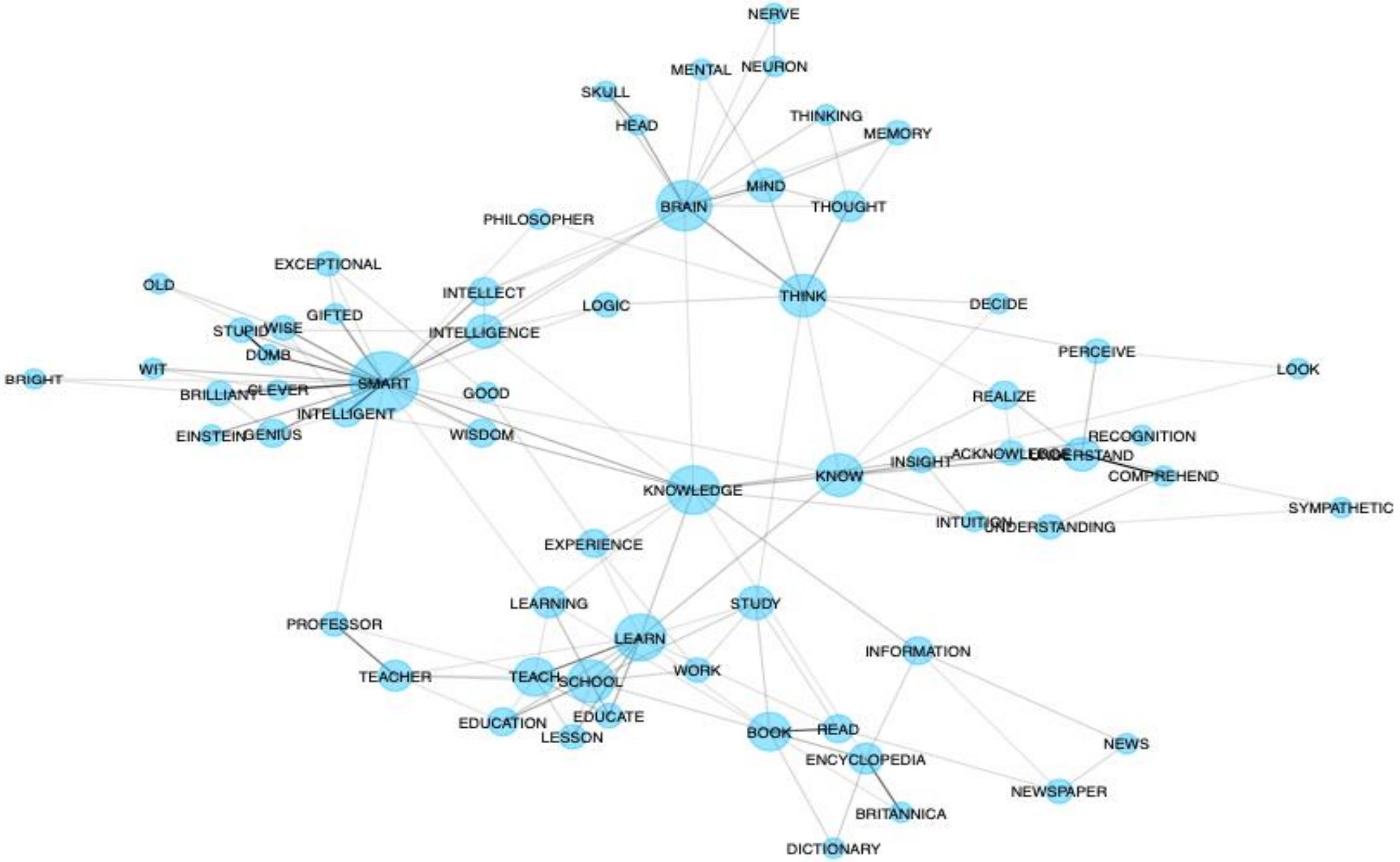
MIND CLUSTERS INTERPRETATION

Cluster No.	Interpretation
1	Getting Knowledge with Experience and Information
2	Where knowledge is probably stored
3	Classification of a person based on the knowedge they have
4	Resources to obtain knowledge
5	Academic and Educational Knowledge
6	Traits of a person having knowldege

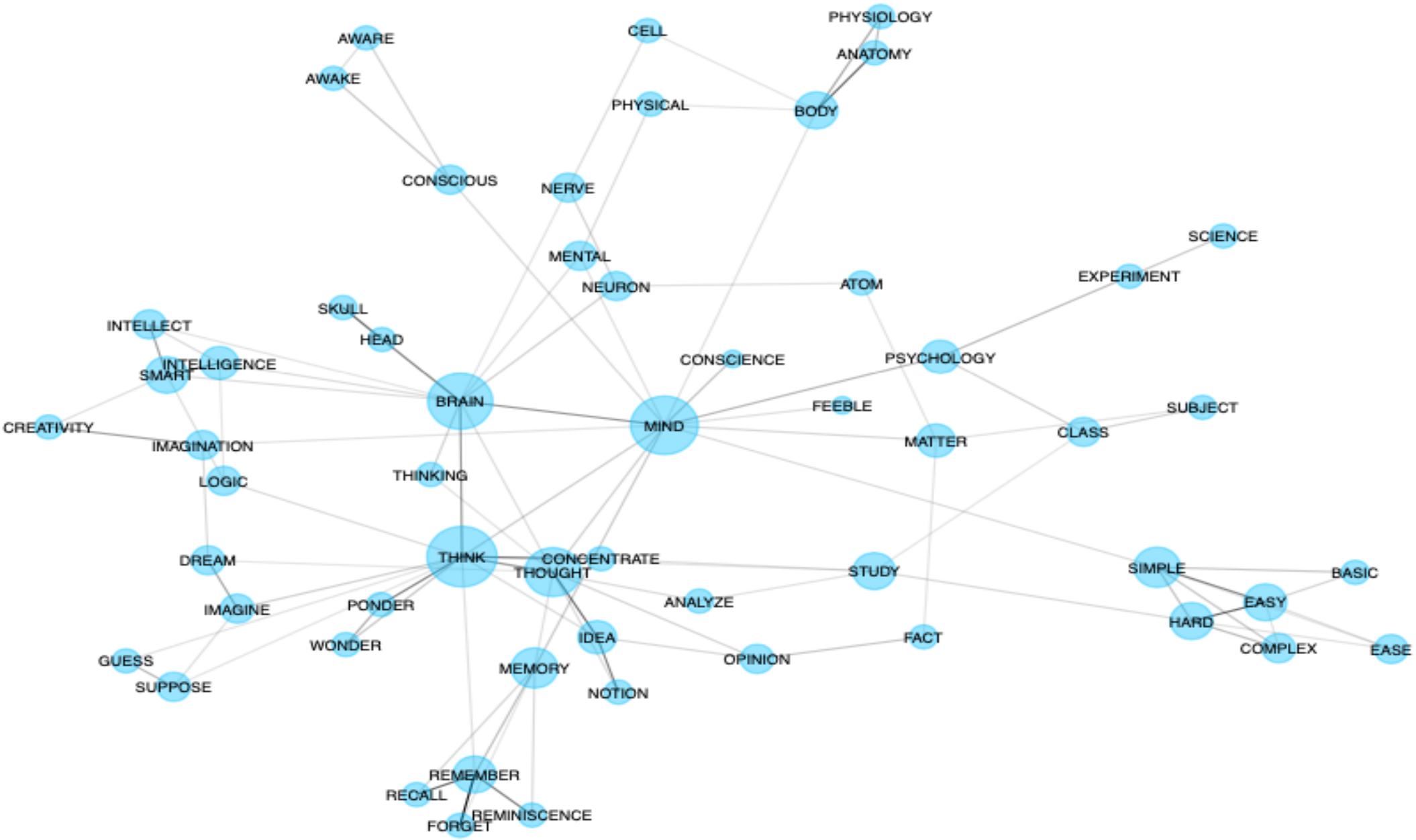
GRAPH PLOTS


ART
DEVELOP
DESTROY
INVENT
MAKE
CREATE
COMPOSE
DESIGN
WRITE
SCRIBBLE
PENCIL
SKETCH
DRAW
DRAWING
DIAGRAM
PICTURE
PORTRAIT
PHOTO
ALBUM
WALL
PLASTER
HOUSE
BRUSH
PAINTER
PAINT
PAINTING
ARTS
ABSTRACT
CRAFTS
ART
CRAFT
MUSEUM
WRITING
SCULPTURE
CREATIVE
CREATIVITY
CANVASS
ARTIST
CANVAS
SHOES
TENT
CREATION
GOD
GODDESS
BEAUTY
GIRL
MATERIAL
TALENT
SKILL
ABILITY

KNOWLEDGE


NERVE
MENTAL
NEURON
SKULL
HEAD
THINKING
MEMORY
BRAIN
MIND
THOUGHT
PHILOSOPHER
EXCEPTIONAL
OLD
INTELLECT
LOGIC
THINK
DECIDE
GIFTED
STUPID
WISE
INTELLIGENCE
DUMB
PERCEIVE
LOOK
WIT
BRIGHT
SMART
BRILLIANT
CLEVER
GOOD
REALIZE
INTELLIGENT
EINSTEIN
GENIUS
WISDOM
RECOGNITION
INSIGHT
KNOW
COMPREHEND
KNOWLEDGE
SYMPATHETIC
INTUITION
UNDERSTANDING
EXPERIENCE
LEARNING
STUDY
PROFESSOR
LEARN
INFORMATION
TEACHER
TEACH
SCHOOL
WORK
EDUCATION
EDUCATE
LESSON
BOOK
READ
NEWS
ENCYCLOPEDIA
NEWSPAPER
BRITANNICA
DICTIONARY

MIND


PHYSIOLOGY
CELL
AWARE
ANATOMY
AWAKE
PHYSICAL
BODY
CONSCIOUS
NERVE
SCIENCE
MENTAL
EXPERIMENT
NEURON
ATOM
SKULL
INTELLECT
HEAD
CONSCIENCE
PSYCHOLOGY
INTELLIGENCE
SMART
BRAIN
FEEBLE
SUBJECT
CREATIVITY
MIND
CLASS
IMAGINATION
MATTER
LOGIC
THINKING
DREAM
THINK
CONCENTRATE
STUDY
SIMPLE
BASIC
THOUGHT
EASY
IMAGINE
PONDER
ANALYZE
HARD
IDEA
FACT
WONDER
COMPLEX
EASE
GUESS
OPINION
MEMORY
SUPPOSE
NOTION
REMEMBER
RECALL
REMINISCENCE
FORGET

REFERENCES:

Kable: https://www.rdocumentation.org/packages/knitr/versions/1.32/topics/kable

Converting .txt file to vector for reading cue.txt: https://stackoverflow.com/questions/23678691/converting-a-text-file-into-a-vector-in-r

Authority score using Page Rank, Degree Centrality, Kleinberg's Method: https://nuigalway.blackboard.com/webapps/blackboard/execute/content/file?cmd=view&content_id=_2331181_1&course_id=_123624_1&framesetWrapped=true

Louvain algorithm: https://nuigalway.blackboard.com/bbcswebdav/pid-2344041-dt-content-rid-21696439_1/courses/2021-CT5113/2021-Community-Detection-Algorithms.html

CODE APPENDIX

```
#importing libraries
library(igraph)
library("scales")
library(ggraph)
library(knitr)
library(tidyr)
library(kableExtra)
#read the graph
g<- read_graph(file="WordPairs.txt",format="pajek")
#make it undirected
g<- as.undirected(g)
#removing the edges pointing to the same node they emerge from
g<-simplify(g)

#reading cue.txt
cueval <- scan("cue.txt", character(), quote = '')
#removing unwanted values
cueval <- cueval[-1:-24]
#making it numeric
cueval <- as.numeric(cueval)
#setting cue attribute to vertex
g <- set_vertex_attr(g, "cueindicator", value = cueval )
length(V(g))
#deleting non cuewords from graph
```

```
g<- delete.vertices(g,which(V(g)$cueindicator == 0)) # remove nodes with degree zero
length(V(g))
#deleteing_vertices with zero degree
g<- delete.vertices(g,which(degree(g)==0))
length(V(g))
#scaling weights from 0 to 1
E(g)$weight<-rescale(E(g)$weight)
#Copying g into three different graphs
gart<-g
gknowledge<-g
gmind<-g

edgedeletion <- function(gvar, wthreshold)
{
  #deleting edges in gvar that are smaller than the threshold
  return(delete.edges(gvar, which(E(gvar)$weight < wthreshold)))
}
#deleting edges
gart<- edgedeletion(gart,0.028)
gknowledge<- edgedeletion(gknowledge,0.026)
gmind<- edgedeletion(gmind, 0.032)
#KNOWLEDGE, MIND, ART
t1<- V(gart)$name %in% c("ART")
t1 <- V(gart)[t1] #Stores node ART
t2<- V(gknowledge)$name %in% c("KNOWLEDGE")
t2 <- V(gknowledge)[t2] #Stores node KNOWLEDGE
```

```r
t3<- V(gmind)$name %in% c("MIND")
t3 <- V(gmind)[t3] #Stores node MIND
#vectors to store random walk result
walk_rand_art<-c()
walk_rand_knowledge<-c()
walk_rand_mind<-c()
#for loop to run random walk 1000 times
for(i in 1:1000)
{
  walk_rand_art <- c(walk_rand_art, random_walk(gart, start= t1, steps=3, stuck = "return"))
  walk_rand_knowledge <- c(walk_rand_knowledge, random_walk(gknowledge, start= t2, steps=3, stuck = "return"))
  walk_rand_mind <- c(walk_rand_mind, random_walk(gmind, start= t3, steps=3, stuck = "return"))
}
vertexdeletion <- function(gvar, walk_rand_var)
{
  vec<- as.numeric(rownames(table(walk_rand_var)))
  # remove nodes not in random walk
  gresult<- delete.vertices(gvar,V(gvar)[which(!(V(gvar)$name %in% V(gvar)[vec]$name))])
  return(gresult)
}
# calling vertexdeletion to delete vertices not in random walk
gart<-vertexdeletion(gart,walk_rand_art)
gknowledge<-vertexdeletion(gknowledge,walk_rand_knowledge)
gmind<-vertexdeletion(gmind,walk_rand_mind)

gart<- delete.vertices(gart,which(degree(gart)==1)) # remove nodes with degree one
```

```r
gknowledge<- delete.vertices(gknowledge,which(degree(gknowledge)==1)) # remove nodes with degree one
gmind<- delete.vertices(gmind,which(degree(gmind)==1)) # remove nodes with degree one
# function to plot graphs
plotgraph <- function(gvar)
{
  gnew<- ggraph(gvar, layout = "fr") +
  geom_edge_link2(aes(edge_alpha = weight),
                  edge_width=0.3, show.legend = FALSE) +
  geom_node_point(aes(size=degree(gvar)),
                  color = "deepskyblue",
                  alpha=0.4, show.legend = FALSE) +
  geom_node_text(aes(label = name), size = 1.75,
                 repel = FALSE) +
  scale_size_area(max_size=10) +
  theme_void()
  return(gnew)
}
# plotting graphs
artplot <- plotgraph(gart)
plot(artplot)
knowledgeplot <- plotgraph(gknowledge)
plot(knowledgeplot)
mindplot <- plotgraph(gmind)
plot(mindplot)
#AUTHORITY_MEASURES
### PAGE RANK FOR ART#####
```

```
alpha = 0.85
pr<- page_rank(gart, algo = "arpack",  directed = TRUE, damping = alpha ,personalized = NULL, options = NULL)
as1<-sort(pr$vector, TRUE)[1:3]
knitr::kable(as1,row.names=TRUE,col.names=c("Authority Score"), "html")
###AUTHORITY SCORE FOR KNOWLEDGE#####
authority<- authority_score(gknowledge, scale = TRUE)$vector
as2<-sort(authority, TRUE)[1:3]
knitr::kable(as2,row.names=TRUE,col.names=c("Authority Score"), "html")
#####DEGREE CENTRALITY FOR MIND #####
degreecent<-degree(gmind)
as3<-sort(degreecent, TRUE)[1:3]
knitr::kable(as3,row.names=TRUE,col.names=c("Authority Score"), "html")
###Louvain algorithm####
clusterdf <- function(gvar)
{
  set.seed(4292)
  communitiesvar<-cluster_louvain(gvar) #Louvain algorithm for community detection
  tmpr<-as.list(membership(communitiesvar)) #Members stored in tmpr as a list
  wordlist<- names(tmpr) #Names(Words) stored in wordlist
  clusternumber <- matrix(0, length(names(tmpr)))
  df <- data.frame(wordlist,clusternumber) #Dataframe having wordlist and clusternumbers (all zeros at this point)
  for( l in names(tmpr))
  {
  rownum<- which(df$wordlist==l) #Getting rownumber which has word l
  df[rownum,2]<-tmpr[[l]] #Storing clusternumber in that rownumber and column 2
  }
```

```
  clusternames<-unique(unlist(as.list(df["clusternumber"]), use.names = FALSE)) #Storing all unique clusternumbers
  newdf <- matrix(ncol = 2, nrow=0)
  newdf <- data.frame(newdf)
  for(s in clusternames) #For each clusternumber
  {
  rowindex<-which(df["clusternumber"]==s) #get the position in dataframe df
  newlist<-c()
  for(r in rowindex) #at all those position we also have word in column word
  {
  newlist<-c(newlist,df[r,1]) #append each word to newlist
  }
  comcount<-length(newlist) #get the count
  newlist<- paste( unlist(newlist), collapse=' ')
  newdf<-rbind(newdf,c(s,newlist,comcount)) #store both words list (newlist) and cluster numbers in newdf
  }
  colnames(newdf)<-c("Cluster Number","Members","Count") #giving column headings
  newdf<-newdf[order(newdf["Cluster Number"]),] #sorting
  return(newdf)
}

artdf<-clusterdf(gart)
knowledgedf<-clusterdf(gknowledge)
minddf<-clusterdf(gmind)
#plotting the tables using kable
knitr::kable(artdf,row.names=FALSE, "html")%>% add_header_above(header= c("ART"=3))
knitr::kable(knowledgedf,row.names=FALSE, "html")%>% add_header_above(header= c("KNOWLEDGE"=3))
```

```
knitr::kable(minddf,row.names=FALSE, "html")%>% add_header_above(header= c("MIND"=3))
```

```
#Interpretation

artint<-c(" Art as a kind of Talent"," Art for Renovation"," Art for Restoration"," Result of Art like House/Artist/Painter etc"," Art Canvas"," Art as in Divine Creation"," Book Art like Sketching/Scribbling"," Digital Art")

aclnum<-seq(1:8)

artint<-data.frame(aclnum,artint)

colnames(artint)<-c("Cluster No.","Interpretation")

knitr::kable(artint,row.names=FALSE,align = "ll", "html")%>% add_header_above(header= c("ART CLUSTERS INTERPRETATION"=2))

knowledgeint<-c("Getting Knowledge with Experience and Information","Where knowledge is probably stored","Classification of a person based on the knowedge they have","Resources to obtain knowledge","Academic and Educational Knowledge","Traits of a person having knowldege")

kclnum<-seq(1:6)

knowledgeint<-data.frame(kclnum,knowledgeint)

colnames(knowledgeint)<-c("Cluster No.","Interpretation")

knitr::kable(knowledgeint,row.names=FALSE,align = "ll", "html")%>% add_header_above(header= c("KNOWLEDGE CLUSTERS INTERPRETATION"=2))

mindint<-c("Something mind does","Comprehending ability of Mind","Part of body where mind if thought to be there","Mind in the context of Science and experiments","Classifying thought process of a person as in creative, intellectual, etc","Mind as in a body part","Ability of mind to recollect or process past information","Something that puts mind to work")

mclnum<-seq(1:8)

mindint<-data.frame(mclnum,mindint)

colnames(mindint)<-c("Cluster No.","Interpretation")

knitr::kable(knowledgeint,row.names=FALSE,align = "ll", "html")%>% add_header_above(header= c("MIND CLUSTERS INTERPRETATION"=2))
```