Word Association Networks

2020-04-13

Contents

1 Introduction

This report aims to publish the results of my experiments in building Word Association networks for 3 cue words - SERVER, CABLE, SAVIOUR. A word association dataset for English words released by University of South Florida is utilized to build topic networks from words in the vicinity of chosen cue words. Authority measures on the topic network reveal heavily associated words. Community detection reveals which set of words are often closely associated with each other in different contexts

2 Part 1

2.1 Data Preparation

The PAJEK format file was used to create a graph. This network is assumed to be undirected. Later,

- The cue words were located from the text file
- Each node in the graph is give a boolean attribute where True indicates that it was a cue word

2.2 Building Topic Networks - Sampling Strategy

As observed in the section titled 'Part 1' in Appendix, random walks are utilized to build topic networks. The sampling strategy is

- From the Target word , record all the vertices on a random walk of length 3 for 500 times
- For each random walk, construct the edge list by pairing up consecutive vertices

- Combine all the edges into one giant edge list
- Create an undirected graph from this edge list

This is the neighborhood of words for each target word

2.3 Graph Authority measures

The authority measures used are : **Normalized Degree Centrality, Eigen value Centrality & Vertex Betweeness**.

- The Degree centrality is normalized by maximum degree
- Eigen centrailty measure is calculated by specifying that this graph is not a directed graph
- Betweeness is also calculated by specifying that this graph is not a directed graph. This measure is also noramlized

2.3.1 SERVER

- The target word 'SERVER' has the highest degree
- PLATE, HOST not only rank high in Degree centrality but also on Eigen value measure
- FOOD, TENNIS are the words which have the highest betweeness scores
- Servers are most often associated with carrying plates of food. Another common association is with tennis players who start a point

Table 1: SERVER - Nodes with Top 3 Authority Measures

Measures	Words	Values
Degree	SERVER	1.00
Degree	PLATE	0.27
Degree	HOST	0.24
Degree	RESTAURANT	0.21
Eigen value	SERVER	1.00
Eigen value	HOST	0.43
Eigen value	PLATE	0.42
Eigen value	RESTAURANT	0.33
Betweeness	SERVER	0.57
Betweeness	FOOD	0.32
Betweeness	TENNIS	0.27
Betweeness	PLATE	0.21

2.3.2 CABLE

- The target word 'CABLE' has the highest degree
- HBO not only ranks high in Degree centrality but also on Eigen value measure
- CAR, TELEVISION are the words which have the highest betweeness scores
- CABLE is most often associated with TV & TV shows production networks. Another common association is with Cars

Table 2: CABLE - Nodes with Top 3 Authority Measures

Measures	Words	Values
Degree	CABLE	1.00
Degree	HBO	0.20
Degree	TELEVISION	0.17
Degree	CONNECT	0.16
Eigen value	CABLE	1.00
Eigen value	HBO	0.63
Eigen value	CONNECT	0.26
Eigen value	TELEVISION	0.26
Betweeness	CABLE	0.84
Betweeness	CAR	0.29
Betweeness	TELEVISION	0.26
Betweeness	BOX	0.23

2.3.3 SAVIOUR

- The target word 'SAVIOUR' has the highest degree
- BIBLE not only ranks high in Degree centrality but also on Eigen value measure
- FOOD, HELP are the words which have the highest betweeness scores
- JESUS CHRIST is the most common association with SAVIOR, deeds of charity like providing food or medical help is also associated with this target word

Table 3: SAVIOR - Nodes with Top 3 Authority Measures

Measures	Words	Values
Degree	SAVIOR	1.00
Degree	BIBLE	0.19
Degree	CHURCH	0.17
Degree	MEMORY	0.16
Eigen value	SAVIOR	1.00
Eigen value	BIBLE	0.34
Eigen value	CHRIST	0.32
Eigen value	JESUS	0.30
Betweeness	SAVIOR	0.77
Betweeness	FOOD	0.26
Betweeness	HELP	0.22
Betweeness	KNOW	0.18

3 Part 2

Figures 1,2,3 show the topic networks for our target words. It is apparent from these that all of our graphs have densely connected sub graphs. As as result I opted to use the walk trap algorithm which is specialized to find communities in this sort of networks

3.1 Community Detection Algorithm - WalkTrap

WalkTrap tries to find densely connected subgraphs, also called communities in a graph via random walks. The idea is that short random walks tend to stay in the same community.

- The membership parameter is set to true. This forces the algorithm to calculate the membership for each vertex corresponding to the highest modularity value. This is later used for analysis.

3.2 Community Analysis

The community detection algorithm allows us to uncover the different contexts in which the same cue word is utilized. An interpretation of the various detected communities is tabulated here. The various contexts uncovered for our target words are as follows

Table 4: SERVER - Communities

Community	Nodes	Size	Context
1	SERVER HOST PARTY FOOD GRIT GIRL WAITER CHEF WAITRESS ORDER HOSTESS CUPCAKE STEWARDESS DINNER SPOIL SUB CAKE WATER TWINKLES HOUSE BUSBOY STAKE SWALLOW CATTLE HELPER PLANE FAMISHED TIP DIET JOB WORK LADY YOGURT GUEST MONEY SPICE COCKTAIL SHOW CORNBEEF TABLES NUT MUTTON BUY WHEAT NICE PIZZA GROCERIES WOMAN INTAKE CHIP SANDWICH SQUASH WORKER PLENTY ROOT WOMEN SERVE GROCERY COOKOUT PEANUTS BOLOGNA YUCK POT	63	A person whose job is to serve food to customers
2	PLATE SPOON TRAY PARSLEY FASHION CAFETERIA NAME ASH HEAVY DISH BRASS SILVER ARMOR CABINET CART CUP CHINA HOME ICE SILVERWARE PORCELAIN TV GROWTH STEEL LICENSE HOLD SAUCER CERAMIC NAPKIN FLAT CARRY FORK BOWL GLASS	34	Utensil used to serve food
3	TENNIS NET MATCH ACE HEADBAND VOLLEYBALL SPORT COURT RACKET SPORTS RACQUET LET RECREATION GOLF CLAY SHOES TABLE DOUBLE BALL PLAYER ACTIVITY PRO RACQUETBALL SET FAULT	25	A tennis player who serves to start the point
4	RESTAURANT COOK BOOTH DINE PUB ITALIAN MCDONALD'S MANAGER OUT EAT DISHWASHER DINER RECOMMEND MEAL RESERVATION SEAFOOD CHINESE CAFE INN	19	Places you would see a waitor in
5	MAID MILK CHAMBER HOUSEKEEPER BLACK CLEANER UNIFORM BLONDE MAIDEN FRENCH HONOR RUBBER SERVICE SERVANT BUTLER DRESS	16	In-house helpers who work for a wealthy family
6	BARTENDER BEER ALCOHOL BARTER FUN LIQUOR MIXER DRINK DRUNK BAR	10	A person serving alcoholic beverages

Table 5: CABLE - Communities

Community	Nodes	Size	Context
1	CABLE HBO CORD WIRE STRING TELEPHONE EXTENSION PLUG PHONE SHARP TAP CUT OUTLET LINE HANGER ANTENNA LIVE PLIERS HARP FENCE FRAY MUSIC BRACES CAGE SPINE BARBED CONDUCTOR SOCKET TWINE ELECTRIC COCAINE LESS ELECTRICITY FIRE	34	A conductor for transmitting electrical power
2	CAR PARKING TOYOTA VEHICLE SATURN SALESMAN BUY FAST CRASH ADJUSTMENT RADIATOR TRUCK HORN PLATES ROAD MAROON PERFORMANCE MONARCH CLAMP FIX WRECK TIRE MOTION CYLINDER THEFT TRANSPORTATION LEMON SMASH DAMAGE VALVE GAUGE DEMOLISH FLUID CART HITCH HIKER CHASE RIDE	38	A wire used to fix car batteries
3	TELEVISION CHANNEL APPLIANCE CAPTION ACTRESS CONTROLS TODAY CHEER PROGRAM VIDEO VIOLENT MASH SERIES VOLUME SET VIOLENCE PRAIRIE REPAIR ENTERTAINMENT ELECTRICIAN FAME COUCH CARTOON ZENITH PRODUCER SITTING COMMERCIAL SHOW REMOTE SPORTS PRODUCTION PUBLIC	32	An organisation that produces TV shows
4	ROPE KNOT BELT STRAND BONDAGE TIGHT PULL CLIMB SKIP NYLON TOW TUG DOPE STRAP ACROBAT BURN BOAT TIE JUMPY BOUND HANG SOAP NOOSE YARN FIBER	25	A thin object that can be looped or tied around
5	BOX SHOE CASE BOOTH KIT LUNCH BALLOT CEREAL EMPTY JUNK PARCEL HAT FUSE CARDBOARD SUITCASE SUGGESTION JACK PUNCH TRUNK WAREHOUSE GIFT PILL TURTLE CRACKER SHADOW PRESENT PACKAGE PACK FRAGILE	29	A square hollow box that can hold things
6	BATTERY OPERATED WATT TOY AA VOLT POWER POSITIVE DEAD RADIO HEART WATCH CHARGE AMP ACID FLASHLIGHT	16	Recharging dead car batteries
7	UNIT METER WHOLE DIVISION CARE STORAGE MATH KITCHEN PART PIECE CONSOLE SPACE MEASUREMENT STORE DIGIT PRICE SECTION SQUARE CHAPTER CIRCLE ONE CLOTHES	22	A thin line that is used to measure length
8	CONNECT BREAK MATCH JOIN TOGETHER HIT DOTS FOUR GAME SEPARATE DISCONNECT TOUCH ATTACH LINK WIRES	15	Establish a link between two entities
9	PRINTER COPIER PRESS BROKEN COPY IBM LASER PUBLISHER XEROX INK PAPER MACHINE TYPEWRITER CANNON LOUD	15	A wire connecting Printers to office computers
10	COMPUTER NETWORK SYSTEM INTERNET ABC STATION NEWS PEOPLE CHAIN COMMUNICATION TEAM	11	A device that aids in digital communication

Table 6: SAVIOUR - Communities

Community	Nodes	Size	Context
1	SAVIOR GOD PERFECT BELIEF ANGEL GUARDIAN JESUS BELIEVE HOLY CHURCH CHRIST BIBLE PROVERB LORD FLIES HEAVEN LOVE SAINT RELIGION REJOICE HOLINESS GLORY BLESSING PATRIOT SUNDAY FAITH WORSHIP ENGLAND LAMB HALO PRIEST RELIGIOUS PRAY DESCEND MARTYR LADY BLESS PRAISE PASSAGE JUDE CHAPEL HYMN ATTEND SOUL SPIRIT SWEAR BISHOP PSALM BELL FAIRY SACRED SCHOOL CHRISTMAS CHRISTIAN SCROLL HARP PRAYER MYTHOLOGY MARY CROSS POWERFUL METHODIST MASTER MARRY FRIEND GODLINESS REBEL SERVICE SON NICHOLAS CATHEDRAL FRIAR MINISTER READ OMNIPOTENT SLUM PETER CHOIR HEAL BAPTIST CARPENTER DEATH PREACHER SHEPHERD CAKE DEVIL PROTESTANT MURDERER SIN OATH SATIN REPENTANCE NEPTUNE CONVENT CONFESS GATHER FAITHFUL KINGDOM RIGHTEOUS REVIVAL WINGS TRUTH SERMON BERNARD PEW SYNAGOGUE HOPE	107	The Biblical saviour - Jesus, religous saviours
2	HELP AIDS PLEASE OPERATOR SOS FAVOR WANTED TREATMENT ASSISTANCE TEND ASK SUPPORT HOSPITAL DESPAIR SHARE CARRY COMFORT LEVER SUGGESTION SHOUT AMBULANCE DIRECT DEFENSE CUE RECOMMENDATION CRISIS DANGER DIRECTIONS CLERK WARN CRY	31	A person providing medical help
3	FOOD PREY KITCHEN POTATO PIE MOUSSE BUY STIR STAKE ONION CHECKERS DIGESTION CORNBEEF CHIP SUB WAFFLES BROIL JUNK BROCCOLI GRIT NUT OATS CRACKERS BRAN FAVORITE LOLLIPOP ROAST COOKING GRACE SUPER FISH PICNIC ROLL OATMEAL DOUGHNUT MCDONALD'S	36	A person who provides food to the needy
4	TASTE TOUCH UNPLEASANT HORRIBLE TONGUE GOOD TANGY SMELL PREFERENCE BITTER CHOCOLATE ACQUIRE TART MILD BEER YUCK MOUTH SIP SENSE BUD EAT SWEET MINT TRY SPICE	25	Types of food one enjoys - associated to savor
5	KNOW GUESS WISE UNKNOWN CONFUSION NO INTELLIGENT FORESIGHT PERCEIVE FACT IMPORTANT ASSUME ASSOCIATE TEACH DEFINE UPDATE SUGGEST SURE ALL INTRODUCE ACKNOWLEDGE INFORM PREDICT MEET	24	words associated with savor by mistake
6	HERO STAR WHITE BRAVADO SUBMARINE BRAVE TRIBUTE SANDWICH PATRIOTIC GREATEST MAN HEROIN LEGION VETERAN LEGEND	15	A person who saves people in distress
7	MEMORY HINDSIGHT BACKGROUND RECOGNITION REMEMBER MEMORIAL REMIND PICTURE BAD BLANK THINK FORGET CEREMONY LOSS ELEPHANT BRAIN PAST RETAIN SPAN	19	Remembering things fondly - associated to savor

4 Graph Plots

The following graphs are an effort in visualizing the topic networks built using random walks. The nodes are colored by the community to which they belong.

4.1 Topic Network - SERVER

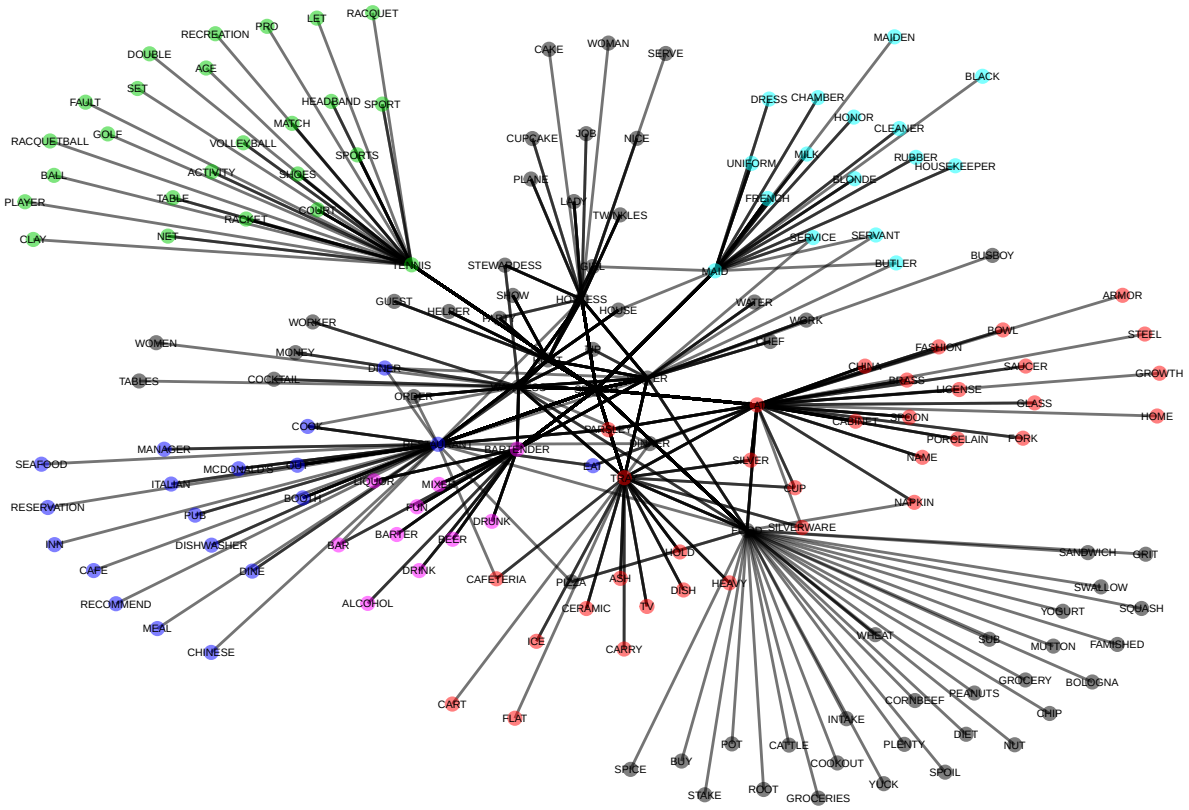

Figure 1: CABLE Topic Network

4.2 Topic Network - CABLE

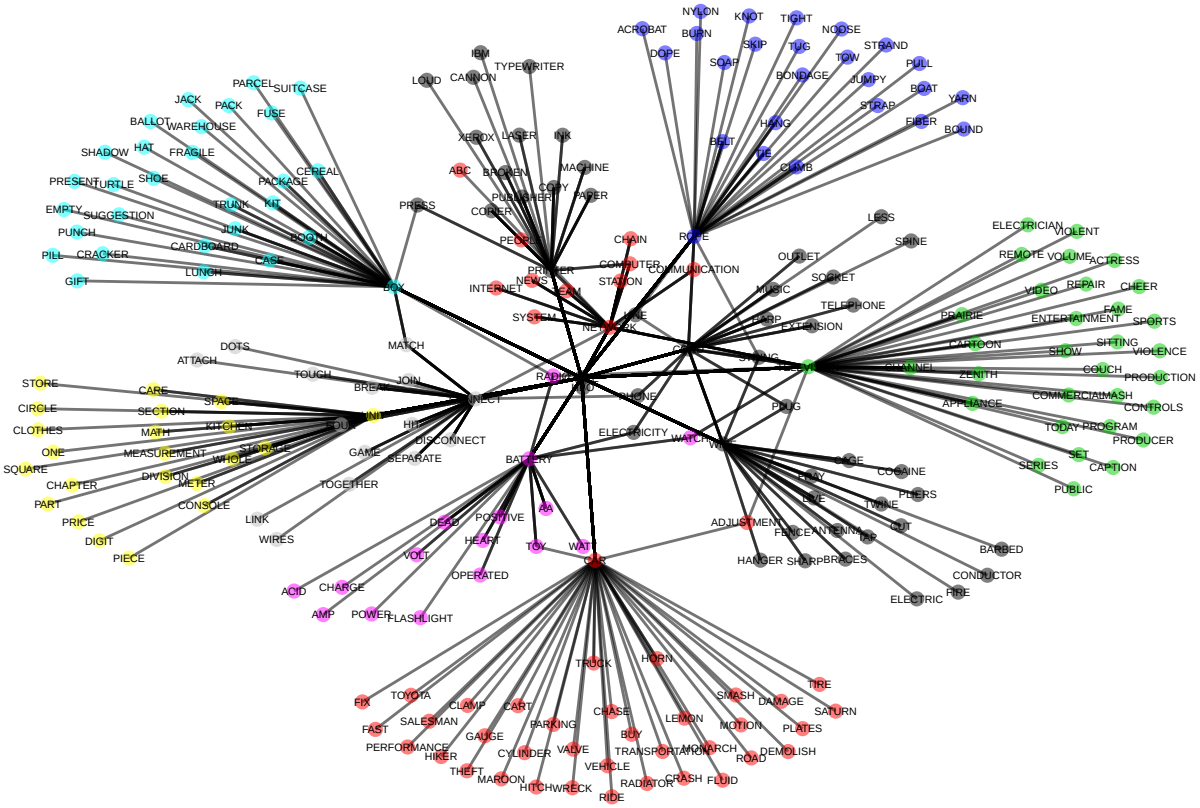

Figure 2: CABLE Topic Network

4.3 Topic Network - SAVIOUR

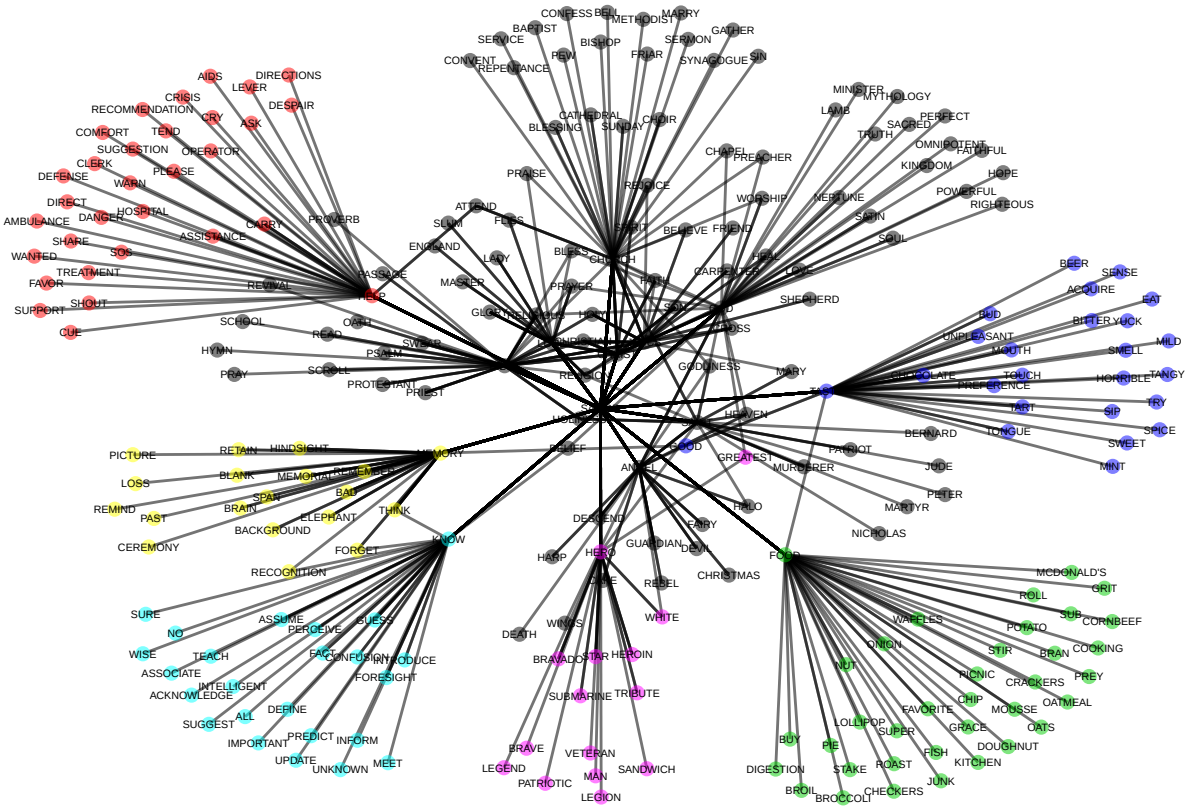

Figure 3: SAVIOUR Topic Network

5 Conclusion

Using the given dataset, random walk sampling method was used to create topic networks for a few target words. These networks were then used to find communities by using the walktrap algorithm. The resulting communities were interpreted to reveal the different contexts the chosen target words are commonly found in.

6 Appendix

```
knitr::opts_chunk$set(echo = FALSE)
library(igraph)
library(ggraph)
library(readr)
library(kableExtra)

#### Part 1 ####

# Load network file
g<- read_graph(file="PairsFSG.net",format="pajek")

# simplify the graph
g<- as.undirected(g)
g<-simplify(g)

#load in data about cue words
cue <- read_lines('cue.txt')[-seq(1,4)]
cue <- ifelse(cue == "1",T,F)

# Mark which of the words in the graph are cue words
V(g)$cue = cue


# Choosing target words
#   - randomly sample 10 cue words
#   - For each word build topic network, find communities in it
#   - Print details of those words who have 6-9 communities
random_targets <- V(g)$name[sample(which(V(g)$cue == T),10)]

for (word in random_targets){

    net <- make_topic_network(g,word)
    net.comm <- cluster_walktrap(net)
    cat(word,"\t",max(net.comm$membership),"\n")

    if (max(net.comm$membership) %in% seq(6,9)){
        print(get_community_nodes(net.comm))
    }


}


target_words = c("SERVER","CABLE","SAVIOUR")

make_topic_network <- function(graph, start_node) {

        # all the vertices on the walk
        walk_vids <-
                lapply(
                        rep(3, 500),
                        random_walk,
```

```
                        graph = g,
                        start = start_node,
                        stuck = "return"
                )

        # create edges by pairing adjacent vertices
        walk_edge_list <-
                lapply(walk_vids, function(v) {
                        named_v <-
                                names(v)
                        c(named_v[1], named_v[2], named_v[2], named_v[3])
                })

        #created an undirected graph from the edge list
        topic_network <- make_undirected_graph(unlist(walk_edge_list))

        topic_network
}



get_community_nodes <- function(comm){

num_comm <- max(comm$membership)

sapply(seq(1,num_comm),function (v) {comm$names[which(comm$membership == v)]})
}

#### Part 1 END ####

set.seed(16)
word1_topic_net <- make_topic_network(g,"SERVER")
word2_topic_net <- make_topic_network(g,"CABLE")
word3_topic_net <- make_topic_network(g,"SAVIOR")


# Returns a 3 member list
#     Member 1 -> Nodes with high degree centrality
#     Member 2 -> Nodes with high eigen value centrality
#     Member 3 -> Nodes with high vertex betweeness
get_authority_nodes <- function(topic_net){

        result = list()

        degree_top3 <- sort(degree(topic_net), decreasing = TRUE)[1:4]
        result$Degree <- degree_top3/max(degree(topic_net))

        eigen_ret <- eigen_centrality(topic_net, directed = F)
        eigen_top3 <-  sort(eigen_ret$vector, decreasing = TRUE)[1:4]
        result$Eigen <- eigen_top3

        degree_top10 <- sort(degree(topic_net), decreasing = TRUE)[1:10]

        betweeness <-
```

```
                sort(betweenness(
                        topic_net,
                        directed = F,
                        normalized = T
                ),
                decreasing = T)[1:4]
        result$Betweeness <- betweeness

        result
}

word1_authorities <- get_authority_nodes(word1_topic_net)
word2_authorities <- get_authority_nodes(word2_topic_net)
word3_authorities <- get_authority_nodes(word3_topic_net)


measures <- c(rep("Degree",4),rep("Eigen value",4),rep("Betweeness",4))
words <- c(
  names(word1_authorities$Degree),
  names(word1_authorities$Eigen),
  names(word1_authorities$Betweeness)
)
values <- c(word1_authorities$Degree,word1_authorities$Eigen,word1_authorities$Betweeness)

data.frame("Measures"= measures,"Words" = words,"Values" = values) %>%
  kable(digits = 2,"latex",caption = "SERVER - Nodes with Top 3 Authority Measures",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped"))


words <- c(
  names(word2_authorities$Degree),
  names(word2_authorities$Eigen),
  names(word2_authorities$Betweeness)
)
values <- c(word2_authorities$Degree,word2_authorities$Eigen,word2_authorities$Betweeness)

data.frame("Measures"= measures,"Words" = words,"Values" = values) %>%
  kable(digits = 2,"latex",caption = "CABLE - Nodes with Top 3 Authority Measures",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped"))

words <- c(
  names(word3_authorities$Degree),
  names(word3_authorities$Eigen),
  names(word3_authorities$Betweeness)
)
values <- c(word3_authorities$Degree,word3_authorities$Eigen,word3_authorities$Betweeness)

data.frame("Measures"= measures,"Words" = words,"Values" = values) %>%
  kable(digits = 2,"latex",caption = "SAVIOR - Nodes with Top 3 Authority Measures",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped"))
set.seed(400)
```

```r
# Community detection using walktrap
word1.comm<- cluster_walktrap(word1_topic_net,  membership = TRUE)

word2.comm<- cluster_walktrap(word2_topic_net,  membership = TRUE)

word3.comm <- cluster_walktrap(word3_topic_net,  membership = TRUE)


all_words <- get_community_nodes(word1.comm)

comm_nodes<- sapply(all_words,function(l) {
    unlist(l)
    paste(l,collapse=' ')
    })

comm_num <- seq(1,length(all_words))
comm_size <- sapply(all_words,length)

contexts <- c(
  "A person whose job is to serve food to customers",
  "Utensil used to serve food",
  "A tennis player who serves to start the point",
  "Places you would see a waitor in",
  "In-house helpers who work for a wealthy family",
  "A person serving alcoholic beverages"
)

data.frame("Community" = comm_num,"Nodes" = comm_nodes,"Size" = comm_size,"Context"= contexts) %>%
kable("latex",caption = "SERVER - Communities",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped")) %>%
    column_spec(2, width = "20em") %>%
    column_spec(4, width = "20em")


all_words <- get_community_nodes(word2.comm)

comm_nodes<- sapply(all_words,function(l) {
    unlist(l)
    paste(l,collapse=' ')
    })

comm_num <- seq(1,length(all_words))
comm_size <- sapply(all_words,length)

contexts <- c(
  "A conductor for transmitting electrical power",
  "A wire used to fix car batteries",
  "An organisation that produces TV shows",
  "A thin object that can be looped or tied around",
  "A square hollow box that can hold things",
  "Recharging dead car batteries",
  "A thin line that is used to measure length",
```

```
  "Establish a link between two entities",
  "A wire connecting Printers to office computers",
  "A device that aids in digital communication"
)

data.frame("Community" = comm_num,"Nodes" = comm_nodes,"Size" = comm_size, "Context" = contexts) %>%
kable(caption = "CABLE - Communities",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped")) %>%
    column_spec(2, width = "25em") %>%
    column_spec(4, width = "15em")


all_words <- get_community_nodes(word3.comm)

comm_nodes<- sapply(all_words,function(l) {
    unlist(l)
    paste(l,collapse=' ')
    })

comm_num <- seq(1,length(all_words))
comm_size <- sapply(all_words,length)

contexts = c(
  "The Biblical saviour - Jesus, religous saviours",
  "A person providing medical help",
  "A person who provides food to the needy",
  "Types of food one enjoys - associated to savor",
  "words associated with savor by mistake",
  "A person who saves people in distress",
  "Remembering things fondly - associated to savor"

)

data.frame("Community" = comm_num,"Nodes" = comm_nodes,"Size" = comm_size,"Context" = contexts) %>%
kable(caption = "SAVIOUR - Communities",
        booktabs = T) %>%
  kable_styling(latex_options = c("striped")) %>%
    column_spec(2, width = "25em") %>%
    column_spec(4, width = "15em")



ggraph(word1_topic_net,layout = "fr") +
        geom_edge_link2(edge_width = 0.5,aes(edge_alpha = 0.1),show.legend = F) +
        geom_node_point(color=as.factor(word1.comm$membership),size=2,alpha = 0.5) +
        geom_node_text(aes(label=name),size=1.5) +
        theme_void()

ggraph(word2_topic_net,layout = "fr") +
        geom_edge_link2(edge_width = 0.5,aes(edge_alpha = 0.1),show.legend = F) +
        geom_node_point(color=as.factor(word2.comm$membership),size=2,alpha = 0.5) +
        geom_node_text(aes(label=name),size=1.5) +
        theme_void()
```

```
ggraph(word3_topic_net,layout = "fr") +
        geom_edge_link2(edge_width = 0.5,aes(edge_alpha = 0.1),show.legend = F) +
        geom_node_point(color=as.factor(word3.comm$membership),size=2,alpha = 0.5) +
        geom_node_text(aes(label=name),size=1.5) +
        theme_void()
```