

Assignment 2 WebnNetwork

Nilay Udeshi

2023-03-23

```
#' A Personalised Page Rank PPR approach to extracting a topic subgraph. The sub  
#' graph is extracted from the topn most high ranked nodes according to the PPR  
#' function based on the target nodes submitted
```

```
#' @param g the graph from which the topic subgraph will be extracted  
#'   calculated.
```

```
#' @param target_node_names a character vector with the  
#' names of the topic nodes
```

```
#' @param topn the number of the top ranked vertices to use  
#' in creating the topic subgraph
```

```
#' @param damping damping factor used in pagerank calculation  
#' the default value is 0.85
```

```
#' @return a new subgraph with topn ranked nodes  
#' according to the PPR score based on the target_node_names submitted
```

```
library(igraph)
```

```
## Warning: package 'igraph' was built under R version 4.1.3
```

```
##
```

```
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      decompose, spectrum
```

```
## The following object is masked from 'package:base':
```

```
##
```

```
##      union
```

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

```
# your code
```

```
  cat("ppr_topic_network has yet to be implemented \n")
```

```
}
```

```

#' A Random walk approach to extracting a topic subgraph. The sub
#' graph is extracted from the most frequently visited nodes in
#' multiple random walks runs on the input set of seed node names.

#' @param g the graph from which the topic subgraph will be extracted
#'   calculated.
#' @param seed_node_names a character vector with the
#'   names of the topic words to seed the random walks
#' @param steps the number of steps each random walk should take
#' @param walks the number of random walks to be made for each topic
#'   names of the topic nodes to seed the random walks
#' @param topn the number of the top ranked vertices to use
#'   in creating the topic subgraph
#' @return a new subgraph from the topn most frequently visited nodes from the random walk

rand_walk_topic_network <- function(g,seed_node_names, steps, walks, mode, topn) {

# your code

cat("rand_walk_topic_network has yet to be implemented \n")

}

```

```

library(igraph)

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
```

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

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

```
## select two word  
## these must be cue words  
target_node_name1 <- "HEART"  
target_node_name2 <- "HEAD"  
  
# 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
```

```
## character vector containing the target words  
## This will be passed as a parameter  
## to the function that finds the word association  
## subgraph surrounding these words  
target_node_names<-c(target_node_name1,target_node_name2)  
  
## depending on which function you implement  
## you will set appropriate values for these  
## parameters  
topn<-0  
steps <- 0  
walks <- 0  
mode <- "all"  
damping = 0  
  
# call ppr function for finding a topic sub graph  
topic_graph<- ppr_topic_network (g,target_node_names, topn, damping )
```

```
## ppr_topic_network has yet to be implemented
```

```
# or
```

```
# call rand_walk function for finding a topic subgraph  
topic_graph<-rand_walk_topic_network(g,target_node_names, steps, walks, mode, topn)
```

```
## rand_walk_topic_network has yet to be implemented
```