Applying Community Detection

Ashwini Ashok Munnolli

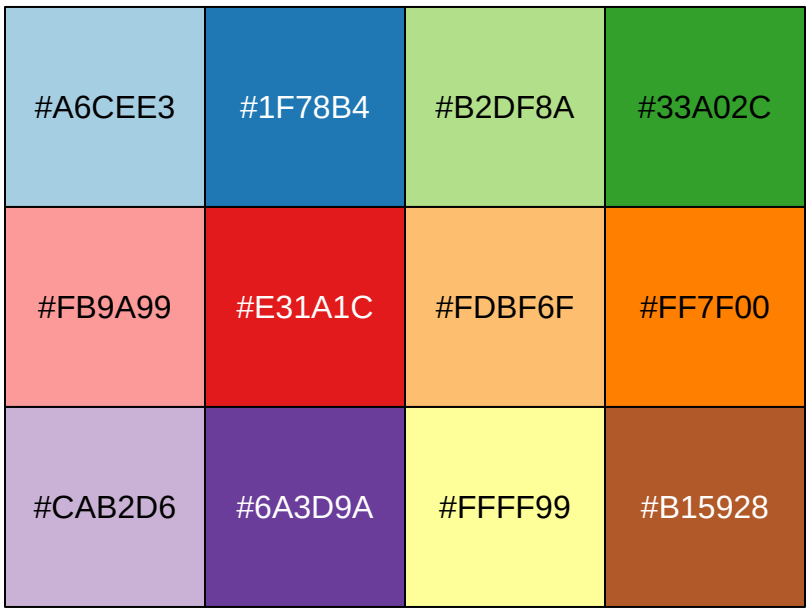
25/08/2022

```
library(igraph)
library(RColorBrewer)
library(scales)


#my_color_pal <- c("#dc661b", "#c9da0b", "#2eb83c", "#157184", "#7a8fe1", "#5525a2","#fd74d8","#b948d5"


#add.alpha <- function(cols, alpha) rgb(t(col2rgb(cols)/255), alpha = alpha)
#colours<-add.alpha(my_color_pal, 0.9)
colours <- brewer.pal(n=12,name = "Paired")

show_col(colours)
```



```r
#code block for single matrix

#check read.table doc skip param
data <- read.table("similarity_matrix_for_graph.csv",sep = ',',header = TRUE, check.names = FALSE, skip

#DF to matrix
data_matrix <- data.matrix(data)
#colnames of the matrix
colnames(data_matrix)
```

```
##  [1] "87153398" "87153401" "91158359" "91253500" "87212280" "89045465"
##  [7] "87153394" "87153396" "87153393" "91303201" "87153395" "89353055"
## [13] "87153399" "91349874" "87153397" "91042911" "91158353" "91303191"
## [19] "91157419" "90079999" "89034877" "90054688" "88019051" "87272402"
## [25] "87214971" "90294859" "87269956" "90045364" "88273558" "87181747"
## [31] "88048748" "91194654" "89300773" "89090775" "88070619" "89210525"
## [37] "87154824" "88105976" "91157909" "87170682" "88093278" "87212275"
## [43] "89090802" "88132177" "89057110" "87266840" "91048168" "87127179"
## [49] "89205573" "91029107"
```

```r
#making all lower triangle values zero in the matrix including diag
data_matrix[lower.tri(data_matrix,diag = TRUE)] <- 0

#number of zeros in the matrix
#length(which(data_matrix == 0))

#avg of non-zero values in the matrix
sum(data_matrix[data_matrix != 0])/length(data_matrix[data_matrix != 0])
```

```
## [1] 0.1903989
```

```r
summary(data_matrix[data_matrix != 0])
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.05502 0.10655 0.18149 0.19040 0.25077 0.62199
```

```r
fivenum(data_matrix[data_matrix != 0])
```

```
## [1] 0.05501601 0.10654775 0.18148821 0.25076880 0.62198535
```

```r
#3rd quadrant
threshold <- fivenum(data_matrix[data_matrix != 0])[4]

#number of values less than threshold (including zeros and lower tria)
length(which(data_matrix >= threshold))
```

```
## [1] 203
```

```
#filter out values below threshold
data_matrix[data_matrix < threshold] <- 0
#data_matrix[data_matrix >= threshold] <- 1
#length(which(data_matrix == 0))
#weighted check
g <- graph_from_adjacency_matrix(data_matrix, weighted=TRUE, mode="upper", diag = FALSE, add.colnames =
#V(g)
#V(g)$medline_ui
#E(g)
#E(g)$weight

summary(g)
```

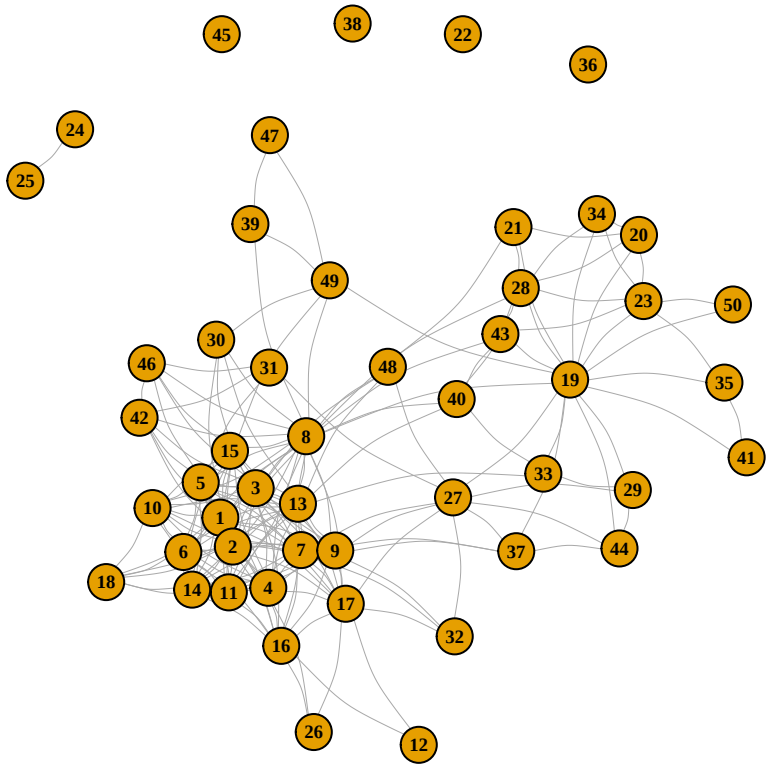
```
## IGRAPH 6c446ec U-W- 50 203 --
## + attr: medline_ui (v/c), weight (e/n)
```

```
#summary(E(g)$weight)

#fivenum(E(g)$weight)[2]


#plot(g)
```

```
vertex_size <- 10
cex_size <-0.6
par(mar=c(0,0,1,0)+.1)
plot( g,
layout=layout_nicely,
vertex.label.cex=cex_size,
vertex.label.color="black",
#vertex.label.dist = 0.7,
vertex.label.font = 2,
vertex.size = vertex_size,
#edge.color = "gray90",
edge.width=0.5,
edge.curved=0.2,
main = paste("Sample graph")
)
```




```
#By default the 'weight' edge attribute is used as weights.Larger edge weights correspond to stronger c
fg.community<- cluster_fast_greedy(g)
table(fg.community$membership)
```

```
##
##  1  2  3  4  5  6  7  8
## 10 17 17  2  1  1  1  1
```

```
#modularity of actual split
modularity(fg.community)
```

```
## [1] 0.3036332
```

```
#modularity of actual split
modularity(g,membership(fg.community),E(g)$weight)
```

```
## [1] 0.3036332
```

```
#plot(as.dendrogram(as.hclust((fg.community))))

#is_hierarchical(fg.community)
fg.10_comm <- cut_at(fg.community,no = 10)
table(fg.10_comm)
```

```
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  2 17  2  6 17  2  1  1  1  1
```

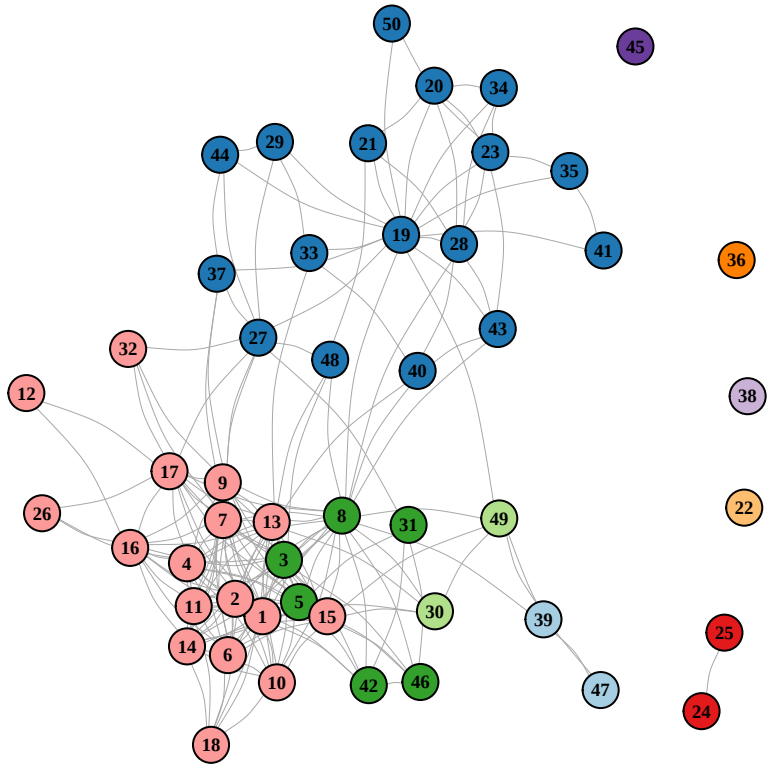
```
#modularity of cut at 10
modularity(g,fg.10_comm,E(g)$weight)
```

```
## [1] 0.2947539
```

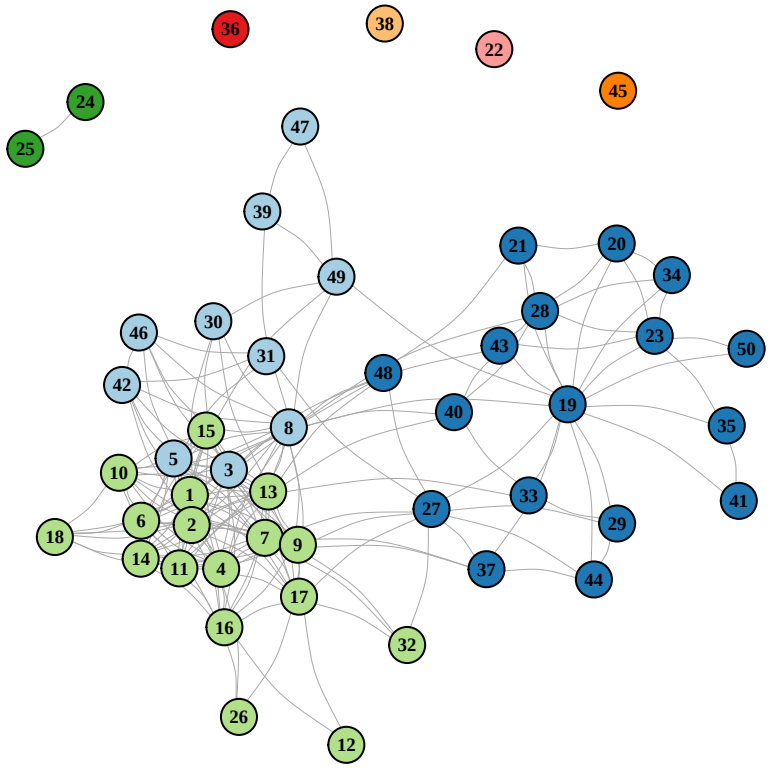
```
#length(fg.10_comm)

#weights: Optional positive weight vector. If the graph has a weight edge attribute, then this is used
#louvain.community<- cluster_louvain(g)
#length(louvain.community)
```

```
par(mar=c(0,0,1,0)+.1)
plot( g,
layout=layout_nicely,
vertex.color=colours[fg.10_comm],
vertex.label.cex=cex_size,
vertex.label.color="black",
#vertex.label.dist = 0.7,
vertex.label.font = 2,
vertex.size = vertex_size,
#edge.color = "gray90",
edge.width=0.5,
edge.curved=0.2,
main = paste("Cut at 10")
)
```




```
par(mar=c(0,0,1,0)+.1)
plot( g,
layout=layout_nicely,
vertex.color=colours[fg.community$membership],
vertex.label.cex=cex_size,
vertex.label.color="black",
#vertex.label.dist = 0.7,
vertex.label.font = 2,
vertex.size = vertex_size,
#edge.color = "gray90",
edge.width=0.5,
edge.curved=0.2,
main = paste("Actual communities")
)
```




```
#code for all 63 queries
# to find clusters and new top 10 docs

actual_clusters =  c()
final_clusters = c()

orig_top_50 <- list()
new_top_10 <- list()


for (i in 1:63) {

data <- read.table("similarity_matrix_for_graph.csv",sep = ',',header = TRUE, check.names = FALSE, skip

#DF to matrix
data_matrix <- data.matrix(data)

#colnames of the matrix
#colnames(data_matrix)

#making all lower triangle values zero in the matrix including diag
data_matrix[lower.tri(data_matrix,diag = TRUE)] <- 0


#3rd quadrant
threshold <- fivenum(data_matrix[data_matrix != 0])[4]
```

```
#filter out values below threshold
data_matrix[data_matrix < threshold] <- 0

#weighted check
g <- graph_from_adjacency_matrix(data_matrix, weighted=TRUE, mode="upper", diag = FALSE, add.colnames =


#summary(g)


fg.community <- cluster_fast_greedy(g)
actual_clusters[i] <- length(fg.community)
#sizes(fg.community)

if(length(fg.community) >= 10)
{
  final_clusters[i] <- length(fg.community)
  print(sizes(fg.community))
  print(fg.community$membership)
  mem_vector <- fg.community$membership
  n_comm <- length(fg.community)
}
else
{
  fg.10_comm <- cut_at(fg.community,no = 10)
  final_clusters[i] <- 10
  print(table(fg.10_comm))
  print(fg.10_comm)
  mem_vector <- fg.10_comm
  n_comm <- 10
}

#initialisations
index <- c(1)
seen_clusters <- c(mem_vector[1])

for (n in 2:50)
{
  if (!(mem_vector[n] %in% seen_clusters))
  {
    seen_clusters <- append(seen_clusters,mem_vector[n])
    index <- append(index,n)
  }

}


#taking only top 10
new_top_10 <- append(new_top_10,list(index[1:10]))
orig_top_50 <- append(orig_top_50,list(V(g)$medline_ui))

}
```

```
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 16 26  1  1  1  1  1  1  1  1
##  [1]  2  2  2  2  1  1  2  2  1  1  1  1  2  1  1  1  1  2  3  2  2  1  2  4  2
## [26]  2  2  2  5  2  6  2  1  1  1  2  2  2  1  7  8  1  2  2  2  2  2  9  2 10
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  5 10 13  6 11  1  1  1  1  1
##  [1]  3  5  4  6  2  2  5  7  8  2  9  5  5  4  5  2  2  1  2  2  5  3  4  3  1
## [26]  3  3  4  3  5  4  3  4  3  1  3  2  3  3  5  2  5  3  1  1  5 10  2  3  5
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 17  2 10 15  1  1  1  1  1  1
##  [1]  1  3  3  1  5  3  3  6  1  4  7  1  1  3  1  1  4  4  1  4  8  3  1  4  4
## [26]  4  4  1  3  9  4  1  3  1  1  1  2  1  4  4  1  3  4  3  1  2  4  4 10  4
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  2 17  2  6 17  2  1  1  1  1
##  [1]  5  5  4  5  4  5  5  4  5  5  5  5  5  5  5  5  5  5  2  2  2  7  2  6  6
## [26]  5  2  2  2  3  4  5  2  2  2  8  2  9  1  2  2  4  2  2 10  4  1  2  3  2
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 19 10  2 13  1  1  1  1  1  1
##  [1]  2  1  1  2  1  4  4  2  2  4  4  4  4  4  2  1  2  1  4  1  5  1  1  1  2
## [26]  1  1  1  2  1  4  4  1  1  3  4  1  1  6  4  7  8  1  9  1 10  2  3  4  2
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 12 10 12  9  2  1  1  1  1  1
##  [1]  5  5  2  3  6  1  3  4  2  3  1  1  2  1  3  1  3  2  2  4  1  3  2  1  1
## [26]  7  4  8  1  4  4  2  9  2  4  1  1  4  1  3  4 10  3  4  2  3  3  3  3  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13
## 19  2  6 14  1  1  1  1  1  1  1  1  1
##  [1]  1  4  1  4  4  1  4  1  4  4  1  4  2  1  1  5  6  3  1  4  4  1  1  4  1
## [26]  1  4  1  1  1  3  1  4  3  1  2  4  7  3  8  9 10 11  3  3  4 12 13  1  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
##  6 10  2  2  4  8  5  2  2  1  1  1  1  1  1  1  1  1
##  [1]  7  1  7  5 10  2  2  1  5  2  6  2  2  7 11  1  2  6 12 13  2 14 15  6  1
## [26]  1  8  1  2  4  4  6  8  6  7  2  6  9  7  6  2  6  3 16 17  3  9  5 18  5
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 16 12 14  2  1  1  1  1  1  1
##  [1]  1  2  3  1  3  1  2  1  3  1  3  3  2  2  1  5  3  2  3  1  1  3  6  4  3
## [26]  2  3  1  1  2  3  1  3  2  1  1  3  4  7  1  2  1  8  2  9 10  1  3  2  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11
## 15 16  2  4  2  6  1  1  1  1  1
##  [1]  1  2  7  2  2  2  2  2  1  1  4  1  2  2  1  2  1  2  3  2  1  1  1  1  4
## [26]  1  4  2  8  6  6  5  6  3  9  1  1  2  2  2  1  2  4  6  6 10  6  5  1 11
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 20 20  3  1  1  1  1  1  1  1
##  [1]  1  1  1  2  1  2  1  4  3  1  1  1  2  2  1  2  3  2  2  1  1  1  3  2  1
```

```
## [26]  2  2  1  1  1  5  2  1  6  1  2  7  1  1  2  2  2  2  2  8  2  9 10  2  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
## 14  9  3  9  2  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  4  2  5  6  4  7  2  2  4  2  1  4  2  4  2  5  4  4  2  4  2  1  8  9  1
## [26] 10 11  4 12  2  1  3 13  1  1  1 14 15 16  1  3  1  3  1  1 17  1 18  1  1
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  4 12  9 10  2  9  1  1  1  1
##  [1]  2  3  6  2  2  6  2  2  2  3  4  3  4  1  4  2  6  6  6  3  6  7  6  1  4
## [26]  4  3  3  2  8  1  9  4  4  3  2  5  1  3  4  5  6  2  4  2  6  3  4 10  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11
## 15 17  3  8  1  1  1  1  1  1  1
##  [1]  3  2  1  1  4  1  4  1  2  1  2  2  3  5  2  1  6  1  7  2  1  8  2  1  4
## [26]  1  4  4  2  4  2  4  9  2  2 10  1  2  2  4  2  1  2  1  2  1 11  1  2  3
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 15 12  2 15  1  1  1  1  1  1
##  [1]  1  4  1  1  5  4  1  1  2  4  2  1  1  2  1  6  4  1  1  1  7  1  3  1  4
## [26]  2  2  8  4  1  9 10  2  2  4  4  2  2  4  4  4  1  4  3  2  4  4  2  2  4
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 11 18  9  4  3  1  1  1  1  1
##  [1]  2  2  5  3  2  1  5  3  2  2  2  2  2  1  6  3  3  2  2  3  4  7  5  3  2
## [26]  1  1  3  8  2  2  4  3  9  1  1  1  1  2  1  2  4  3  1  4  2 10  2  1  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10
## 15 12  6  6  6  1  1  1  1  1
##  [1]  3  1  1  2  1  1  1  2  1  5  1  1  1  5  2  2  2  5  3  4  6  5  4  2  1
## [26]  3  2  2  7  4  1  1  8  3  5  2  5  2  4  2  2  1  1  4  1  9  3  4  3 10
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 14  2 16 11  2  1  1  1  1  1
##  [1]  3  1  3  5  3  3  3  3  1  3  1  4  3  3  6  7  3  4  1  8  5  1  3  9  1
## [26]  3  4  3  4 10  1  1  1  1  1  4  1  3  4  4  4  1  4  4  3  4  2  3  1  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19
## 11  7  4 11  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  4  4  4  4  4  4  4  4  6  4  4  4  7  8  5  2  3  3  5  9 10 11  2  1  1
## [26]  2  1  1 12  1  1  5  2 13 14  2 15  1 16 17 18  1  2 19  1  1  3  1  3  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11
## 21 12  5  4  2  1  1  1  1  1  1
##  [1]  2  2  1  1  6  2  1  2  1  2  2  7  1  3  1  8  1  1  1  1  1  2  1  9 10
## [26]  1  1  1 11  2  2  2  2  4  1  1  1  3  1  5  3  1  3  3  2  5  4  4  4  1
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 18  4  2 19  2  1  1  1  1  1
##  [1]  4  4  1  4  6  4  4  4  4  2  4  1  7  4  1  8  4  4  1  1  4  1  3  9  1
## [26]  4  4  5 10  1  1  4  2  4  3  1  4  1  1  2  1  1  4  1  1  2  1  4  5  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  8  7  2 12  3  3  2  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
##  [1]  2  4  4  8  1  4  9 10  2  4  4  7 11  6  4  4  4  2  2  1  1 12  1  2  2
## [26]  3  7  6 13 14 15 16  1  5 17 18  4  1  1  6  4 19  4 20  2  5  3  4  1  5
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12
##  9  9  2  2 10 12  1  1  1  1  1  1
##  [1]  4  5  5  6  2  2  7  6  8  9  6  1  5  6  6  5  2  5  2  1  5  1  6  4  2
## [26]  1  6  6 10  1  2  1  1  1  5  3  5  6  2  2  6 11  6  5 12  5  6  2  1  3
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13
## 16 10  3  9  4  1  1  1  1  1  1  1  1
##  [1]  1  1  1  1  1  6  7  8  5  3  1  1  2  2  9  2  4  1  4  1  4  1  3  4 10
## [26]  4  4  1  1  1  4 11  5  2 12  1  1  1  4  2  4  3  2  2  2  2  2  5  5 13
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13
## 19  6  2  5  3  8  1  1  1  1  1  1  1
##  [1]  1  1  6  2  6  2  1  4  1  1  7  1  2  1  1  1  8  2  6  1  5  1  1  2  1
## [26]  5  6  1  5  9  6 10 11  1  4  1  1  2 12  3  1  4  4  6  3  4  6 13  1  6
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  2 10 19 13  1  1  1  1  1  1
##  [1]  4  4  4  2  4  2  2  2  2  2  2  4  4  2  2  4  2  3  3  3  3  4  4  3  4
## [26]  3  3  4  5  3  4  3  6  3  4  3  7  1  1  3  8  9  3  3  3  3  3  3  3 10
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
## 10  7  3  2  4  3  3  4  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  2  2  1  9  1 10  7  3 11  1 12 13  6 14 15  6  2  5  5  1  5  2  3 16  2
## [26]  8  2 17 18 19  8  1  1  8  1 20  7  2  4  3  7 21  1 22  6  4  8  1  5  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
##  8  6  7  2  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
## 27 28 29 30
##  1  1  1  1
##  [1]  3  3  1  6  3  7  8  9 10 11 12 13  4 14  1  5 15 16 17 18 19  1  2  2 20
## [26]  3  5 21  3  2  1 22 23  2 24  1 25 26  1 27 28  1  4  3  1  2 29  3 30  2
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  5 17 21  1  1  1  1  1  1  1
##  [1]  3  1  2  3  3  4  1  5  2  6  2  3  1  2  3  7  3  2  3  2  2  2  3  3  2
## [26]  3  3  3  3  3  2  2  3  3  2  3  3  2  2  3  1  2  2  1  3  3  2  8  9 10
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  9  3 16 16  1  1  1  1  1  1
##  [1]  1  3  4  2  4  4  4  4  3  2  3  5  4  4  3  4  3  3  1  4  4  4  3  1  4
## [26]  6  2  7  4  3  8  4  1  3  9  3  4  3  1  3  1 10  1  3  3  3  4  3  1  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
## 15 13  5  2  5  1  1  1  1  1  1  1  1  1  1
##  [1]  1  2  2  6  1  5  5  1  5  1  1  1  5  1  7  8  9  2  3 10  2  4  3  3  3
## [26]  2  5  1  1  3  4 11  2  2  1  2  1  1 12  2 13 14  1  2  2  1 15  1  2  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14
##  9 10  5  4  6  2  7  1  1  1  1  1  1  1
##  [1]  2  3  2  1  1  1  5  7  6  1  4  7  5  7  3  2  2  1  8  5  3  2  3  2  9
## [26]  1  4  7  1  7  4  2  3  2  5 10  1  2  5  7  5  1 11  6 12  4  2 13  7 14
```

```
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
## 11  5  8 11  2  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  4  4  5  2  1  6  1  1  1  1  2  2  7  4  2  4  8  9 10  3  3  3 11  3  1
## [26] 12  1  5  1 13 14  4  4  4  3 15  1  2  4  4  4  3  3 16  4 17 18  1  1  3
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 18  7  2  2  4 13  1  1  1  1
##  [1]  1  6  6  5  6  6  6  6  6  5  1  1  4  2  1  6  6  5  2  1  7  1  1  1  2
## [26]  1  1  8  5  2  1  6  1  1  2  1  1  1  1  2  1  6  6  3  3  9  2  4  6 10
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13
## 12  5  7 17  1  1  1  1  1  1  1  1  1
##  [1]  4  1  3  1  2  4  2  4  2  4  4  4  1  4  3  1  1  1  3  1  4  3  5  1  3
## [26]  4  4  1  4  6  4  3  7  1  8  3  4  4  1  4  4  9  1 10  2 11  2  4 12 13
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  8  7  8  9 13  1  1  1  1  1
##  [1]  1  1  1  3  2  1  1  2  2  5  6  5  5  4  1  4  1  4  2  5  2  1  5  5  3
## [26]  3  7  5  8  4  4  4  3  5  4  9  5 10  5  5  4  3  5  3  5  3  4  2  3  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 15  5  2 11  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  3  1  6  2  5  5  1  7  3  1  8  1  1  1  9  4 10 11 12 13  1  1  1 14  4
## [26]  1  4  1 15 16  4  1  1  4  4  4 17  4 18  2  1  4 19  4  1  2  2 20  4  2
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  2 17 19  6  1  1  1  1  1  1
##  [1]  2  3  3  3  3  2  4  2  2  2  2  2  2  2  4  3  3  4  2  2  5  2  4  1  2
## [26]  4  4  3  3  3  3  6  3  3  3  3  1  7  8  3  9  2 10  3  2  3  2  2  3  3
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  7  9  5  8  5  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  1  4  5  6  1  3  4  4  7  5  2  3  8  2  5  9  2 10 11  1  2  4  5  1  3
## [26] 12  4  2  1  4 13 14  2 15  1  5 16 17  3  4  2  4 18 19  3  1  2 20 21  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
## 14  4  8  2  9  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  5  1  5  5  6  1  5  3  3  5  1  1  7  1  3  1  2  1  1  5  1  2  3  5  1
## [26]  1  3  3  8  5  1  1  2  3  9 10  1  3 11 12  4 13 14  5 15  4 16  2 17 18
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14
## 13  9  5 13  1  1  1  1  1  1  1  1  1  1
##  [1]  4  2  1  4  2  4  4  1  3  1  1  2  5  6  4  2  1  3  1  2  7  3  8  2  9
## [26]  4  1  2 10  1  1  4  3  4  2 11  1  4  4  1  2  1  3  4 12  1 13 14  4  4
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11
##  6  7  8  9 14  1  1  1  1  1  1
##  [1]  1  3  3  4  2  4  3  5  5  5  3  1  4  5  5  6  2  2  5  5  5  4  3  3  2
## [26]  7  1  8  5  5  9  4  2  4  3  4  2  1 10  5  2  4  3  4  5  1  5  1 11  5
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 19  8 16  1  1  1  1  1  1  1
##  [1]  3  1  3  1  1  1  3  1  3  3  3  3  1  3  3  3  3  3  2  1  1  3  4  1  1
```

```
## [26]  3  3  2  1  5  1  2  1  1  2  6  1  1  1  2  3  2  7  8  1  2  1  9  2 10
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  9 13  2 15  2  2  3  2  1  1
##  [1]  7  3  2  6  1  2  7  8  7  4  4  2  3  4  8  4  2  9  1  2  4  5  1  4  2
## [26]  4  1  2  4  4  1  2  4  2  1  6  4  5  1  4  4  2 10  2  4  2  4  1  1  2
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 13 10  8  2  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  2  2  2  2  2  1  3  2  2  2  1  3  2  1  2  3  6  1  4  1  7  8  1  1  9
## [26]  3  3  1  3 10 11  5  1 12  1  1  1  5 13  4 14 15  1  3 16  3 17 18 19 20
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
## 13  8 15  2  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  1  1  1  2  5  2  6  3  3  3  1  4  3  1  3  2  3  3  7  8  3  9  2 10  4
## [26]  1  3  2 11 12  1  3  2  3  1  3  2 13 14  1  1  3  3  2 15  3  1  1 16  1
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 10  4 17  2 12  1  1  1  1  1
##  [1]  3  3  3  2  3  3  3  3  3  5  1  5  3  5  3  1  1  3  2  5  1  3  4  1  1
## [26]  1  1  1  5  5  4  2  2  5  5  6  5  1  3  3  3  7  3  5  8  5  9  3  5 10
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 15  9  3  6 10  2  2  1  1  1
##  [1]  3  7  7  1  4  4  4  5  4  3  1  1  2  2  1  1  1  4  2  1  1  1  5  5  2
## [26]  5  5  1  2  3  2  5  4  8  1  1  9  2  5  2  5  1 10  5  2  1  6  1  5  6
## Community sizes
##  1  2  3  4  5  6  7  8  9 10
## 11  5  4 13  2 11  1  1  1  1
##  [1]  1  6  5  6  4  4  3  2  6  4  6  1  6  1  4  6  6  1  7  1  4  5  4  4  2
## [26]  2  8  1  4  6  4  6  2  6  3  1  1  3  4  3  1  9 10  2  6  4  4  1  1  4
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 15  7 13  9  1  1  1  1  1  1
##  [1]  1  4  2  1  5  2  1  1  2  2  1  4  1  1  3  3  3  6  1  1  4  1  3  3  1
## [26]  3  3  4  2  1  7  8  3  4  2  1  4  3  9  2  4  3  3  1  3  4  4  1  3 10
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
##  8  3 12  5 17  1  1  1  1  1
##  [1]  3  3  3  6  4  5  3  5  5  3  3  3  5  5  5  1  1  1  5  5  1  5  5  5  1
## [26]  3  2  4  5  7  5  5  1  3  2  4  4  1  3  5  4  8  5  3  9  2  3 10  1  5
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 18  4 15  7  1  1  1  1  1  1
##  [1]  1  4  2  4  1  1  3  1  2  5  1  6  1  1  7  1  8  3  2  3  1  3  3  2  1
## [26]  1  1  1  3  4  3  3  1  3  1  3  4  1  1  4  4  3  1  3  3  9  3  3 10  4
## Community sizes
##  1  2  3  4  5  6  7  8  9 10
## 17  2 12 13  1  1  1  1  1  1
##  [1]  5  3  3  6  7  3  3  1  3  1  1  1  4  4  1  1  1  4  4  1  4  4  1  4  3
## [26]  3  8  4  1  4  4  3  1  1  4  3  1  9  1  2  4 10  3  3  1  3  2  1  4  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
##  9  4  5  3  7  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
```

```
##  [1]  1  1  3  3  1  1  2  3  7  3  3  5  1  5  2  8  4  1  9  1 10  6  5  4 11
## [26]  4 12 13 14 15 16  5  6 17  5 18 19  6 20  5 21 22  5  2  1 23 24  1  2 25
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##  7  5  6  6  3  4  2  4  2  1  1  1  1  1  1  1  1  1  1  1
##  [1]  7  4  1  1  3 10  3  6  1  4  8 11  1  4  5  2 12  1  9  5 13  4  6  5  2
## [26] 14 15  2  1  8 16  6  2  4  4  8  3 17 18  6  7  8  3  9  2  3  3 19 20  1
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 14  4 10  5 12  1  1  1  1  1
##  [1]  3  5  1  5  3  5  5  3  2  2  4  5  1  1  1  5  3  1  1  4  6  3  2  3  3
## [26]  1  5  1  7  3  1  4  5  4  5  8  1  1  1  9  1  4  3  1  3  5 10  5  2  5
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11
## 10 13 11  9  1  1  1  1  1  1  1
##  [1]  5  1  3  2  6  1  3  1  2  2  3  3  4  3  1  2  4  3  4  2  1  2  4  1  7
## [26]  3  2  8  9  4  3  4  2  4  2 10  4  2  3  4  3  1  2  1  2  1  3  2 11  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
##  9  5 11  5  5  2  2  1  1  1  1  1  1  1  1  1  1  1
##  [1]  8  1  5  1  5  6  6  4  7  3  9  3 10  7  1 11  4  3  1  3  2  1  5  3  1
## [26]  3 12  3  3 13  5 14  1  2  3  3  4 15 16  2  4 17  2  2  3 18  4  1  5  1
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 17  7  7 13  1  1  1  1  1  1
##  [1]  1  4  1  1  2  5  1  3  4  4  1  6  1  3  1  1  1  1  2  4  1  7  8  2  3
## [26]  3  4  1  1  3  4  9  4  1  2  2  4  4  4  4  2  1  1  3  4  3  2 10  4  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21
##  9  7  8  4  4  3  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  7  2  5  2  5  5  8  5  9 10  6  3  3  3  1  1 11 12  2  4  2  1  4  2  6
## [26]  1  3 13  1 14  6 15  2  1  2 16  3  1 17  1 18  3 19 20  1  4  4  3 21  3
## fg.10_comm
##  1  2  3  4  5  6  7  8  9 10
## 16 14  9  3  2  2  1  1  1  1
##  [1]  4  4  2  2  3  2  2  3  2  2  5  3  6  2  1  1  2  2  3  2  3  2  1  1  4
## [26]  1  1  7  2  6  1  8  1  1  1  5  2  3  9  1  3  3  2  1  1  1  3 10  1  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
## 11  6 15  2  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  5  1  6  1  3  3  3  3  1  1  1  3  3  3  3  1  3  2  3  1  7  8  9  2  2
## [26]  1  1  4 10  4  3 11  2  3  1 12  3  3 13 14 15  2 16  3  2 17 18 19 20  1
## Community sizes
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
## 11  9  5  4  7  2  1  1  1  1  1  1  1  1  1  1  1  1
##  [1]  2  1  1  2  5  2  1  5  7  3  2  4  5  2  1  1  5  1  1  8  5  2  1  5  9
## [26] 10  4 11 12  1  5  2  2 13 14  4  2  6 15  4  1  3 16  3  1  3  3 17 18  6
```

```
length(which(actual_clusters <10))
```

```
## [1] 29
```

```
actual_clusters
```

```
##  [1]  9  8  3  8  7  9 13 18  8 11  8 18  6 11  8  8 10  7 19 11  4 20 12 13 13
## [26]  6 22 30  4  5 15 14 18  7 13  7 20  8 21 18 14 11  5  8 20 16  5  8 10  5
## [51]  9  6 10 25 20  7 11 18  6 21  9 20 18
```

```
mean(actual_clusters)
```

```
## [1] 11.93651
```

```
final_clusters
```

```
##  [1] 10 10 10 10 10 10 13 18 10 11 10 18 10 11 10 10 10 10 19 11 10 20 12 13 13
## [26] 10 22 30 10 10 15 14 18 10 13 10 20 10 21 18 14 11 10 10 20 16 10 10 10 10
## [51] 10 10 10 25 20 10 11 18 10 21 10 20 18
```

```
#read rel_results of top k from file
rel_results <- read.table("relevant_doc_list_top_50_for_63q.csv",sep = ',',header = FALSE, nrows = 63)

#rel_results
new_rel_list_10 = list()
new_rel_df_10 <- data.frame()
for (j in 1:63)
{
  rel_vec <- c()
  for (k in 1:10)
  {
    if (new_top_10[[j]][k] %in% (which(rel_results[j,] == 1)))
    {
      rel_vec <- append(rel_vec,1)
    }
    else
    {
      rel_vec <- append(rel_vec,0)
    }
  }
  new_rel_list_10 <- append(new_rel_list_10,list(rel_vec))
  new_rel_df_10 <- rbind(new_rel_df_10,rel_vec)
}


# write new rel_list to file for jupyter to read
write.table(new_rel_df_10,'new_relevant_docs_top_10.csv',sep =',',row.names = FALSE,col.names = FALSE)
```

```
#Fetching similarity scores before and after

#new_top_10
#orig_top_50
sum_old = 0
sum_new = 0
for (i in 1:63) {
```

```r
sim_matrix <- read.table("similarity_matrix_for_graph.csv",sep = ',',header = TRUE, check.names = FALSE

#DF to matrix
data_matrix_k50 <- data.matrix(sim_matrix)
#print(data_matrix_k50)


#old sim matrix
#first subtract 10 because we dont want similarity if i,i
print(i)
print((sum(data_matrix_k50[1:10,1:10]) - 10) / 90)


#new sim matrix
print((sum(data_matrix_k50[new_top_10[[i]],new_top_10[[i]]]) - 10)/ 90)
if ((sum(data_matrix_k50[1:10,1:10]) - 10)/ 90 < (sum(data_matrix_k50[new_top_10[[i]],new_top_10[[i]]])
{
  print("Similarity score did not decrease")
}


sum_old <- sum_old + (sum(data_matrix_k50[1:10,1:10]) - 10)/ 90

sum_new <- sum_new + (sum(data_matrix_k50[new_top_10[[i]],new_top_10[[i]]]) - 10)/ 90

}
```

```
## [1] 1
## [1] 0.2563354
## [1] 0.0837172
## [1] 2
## [1] 0.2460477
## [1] 0.2379253
## [1] 3
## [1] 0.2443066
## [1] 0.1699285
## [1] 4
## [1] 0.3706662
## [1] 0.08869959
## [1] 5
## [1] 0.4366277
## [1] 0.2413287
## [1] 6
## [1] 0.0729461
## [1] 0.03478602
## [1] 7
## [1] 0.2334765
## [1] 0.03366682
## [1] 8
## [1] 0.05842281
## [1] 0.02276489
## [1] 9
## [1] 0.1897575
```

```
## [1] 0.07015433
## [1] 10
## [1] 0.1861767
## [1] 0.08523318
## [1] 11
## [1] 0.2123987
## [1] 0.09005878
## [1] 12
## [1] 0.101768
## [1] 0.01952708
## [1] 13
## [1] 0.2317843
## [1] 0.1729311
## [1] 14
## [1] 0.1643413
## [1] 0.06887271
## [1] 15
## [1] 0.2125598
## [1] 0.1210474
## [1] 16
## [1] 0.2889223
## [1] 0.2241185
## [1] 17
## [1] 0.3434223
## [1] 0.1437698
## [1] 18
## [1] 0.2678921
## [1] 0.1297733
## [1] 19
## [1] 0.1531355
## [1] 0.02233494
## [1] 20
## [1] 0.2445565
## [1] 0.05713545
## [1] 21
## [1] 0.2037441
## [1] 0.1300189
## [1] 22
## [1] 0.1166815
## [1] 0.05890232
## [1] 23
## [1] 0.1262211
## [1] 0.1520665
## [1] "Similarity score did not decrease"
## [1] 24
## [1] 0.1314883
## [1] 0.07952669
## [1] 25
## [1] 0.1410797
## [1] 0.02416795
## [1] 26
## [1] 0.2859148
## [1] 0.1005429
## [1] 27
```

```
## [1] 0.06948585
## [1] 0.02113052
## [1] 28
## [1] 0.06776753
## [1] 0.04351239
## [1] 29
## [1] 0.1672062
## [1] 0.150239
## [1] 30
## [1] 0.370146
## [1] 0.189698
## [1] 31
## [1] 0.1274758
## [1] 0.03818007
## [1] 32
## [1] 0.1024267
## [1] 0.04696082
## [1] 33
## [1] 0.1317069
## [1] 0.02822534
## [1] 34
## [1] 0.1850538
## [1] 0.07935507
## [1] 35
## [1] 0.179695
## [1] 0.06774597
## [1] 36
## [1] 0.2813692
## [1] 0.1664044
## [1] 37
## [1] 0.05740431
## [1] 0.03020277
## [1] 38
## [1] 0.259178
## [1] 0.07015495
## [1] 39
## [1] 0.04919711
## [1] 0.01443058
## [1] 40
## [1] 0.206638
## [1] 0.0708872
## [1] 41
## [1] 0.1517131
## [1] 0.06421687
## [1] 42
## [1] 0.1115283
## [1] 0.05800115
## [1] 43
## [1] 0.4072721
## [1] 0.2634537
## [1] 44
## [1] 0.1898762
## [1] 0.1347691
## [1] 45
```

```
## [1] 0.2612499
## [1] 0.0334909
## [1] 46
## [1] 0.1620739
## [1] 0.07055952
## [1] 47
## [1] 0.3239874
## [1] 0.1976758
## [1] 48
## [1] 0.1366266
## [1] 0.1191719
## [1] 49
## [1] 0.1939679
## [1] 0.106183
## [1] 50
## [1] 0.1793986
## [1] 0.0918093
## [1] 51
## [1] 0.248814
## [1] 0.120699
## [1] 52
## [1] 0.2740029
## [1] 0.1923383
## [1] 53
## [1] 0.149835
## [1] 0.07213528
## [1] 54
## [1] 0.2216449
## [1] 0.0578513
## [1] 55
## [1] 0.02111827
## [1] 0.004046148
## [1] 56
## [1] 0.163082
## [1] 0.06020714
## [1] 57
## [1] 0.2648054
## [1] 0.1493191
## [1] 58
## [1] 0.018704
## [1] 0.009513986
## [1] 59
## [1] 0.2399083
## [1] 0.1268516
## [1] 60
## [1] 0.03718176
## [1] 0.02278422
## [1] 61
## [1] 0.1444798
## [1] 0.0314394
## [1] 62
## [1] 0.1719409
## [1] 0.05146471
## [1] 63
```

```
## [1] 0.1217586
## [1] 0.03119817
```

```
#mean of (avg (sim score for every i,j))
print("Before and after")
```

```
## [1] "Before and after"
```

```
sum_old/63
```

```
## [1] 0.1900062
```

```
sum_new/63
```

```
## [1] 0.09125882
```