Mapping All of South Park

by Fred

"All characters and events in this show –even those based on real people– are entirely fictional. All celebrity voices are impersonated ... poorly. The following program contains coarse language and due to its content it should not be viewed by anyone."

Step 1: Collect Underpants

We clean the set following the below steps:

1. Unnest tokens (Separate blocks of text into single words)
2. Remove Stop Words
3. Regroup text by Episode

Step 2: ?

We create a document term matrix using the package `text2vec` and use their `sim2()` with `method = "jaccard"` and `norm = "none"`

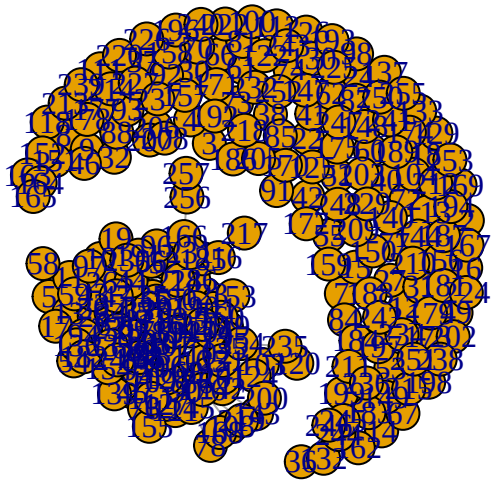
The `sim2()` call gives us a Jaccard Similarity, which returns a value of 1 when two sets are the same.

$$J_{\text{Simmilarity}} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

We instead want the Jaccard Distance, where two equal sets have a value of 0.

$$J_{\text{Distance}} = 1 - \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

From this matrix of distances we can use our very own `FredsVietorisRips` R package to create an adjacency matrix. Using that adjacency matrix and the R package `igraph` we can `graph_from_adjacency_matrix` for visualizations.



Step 3: Profit

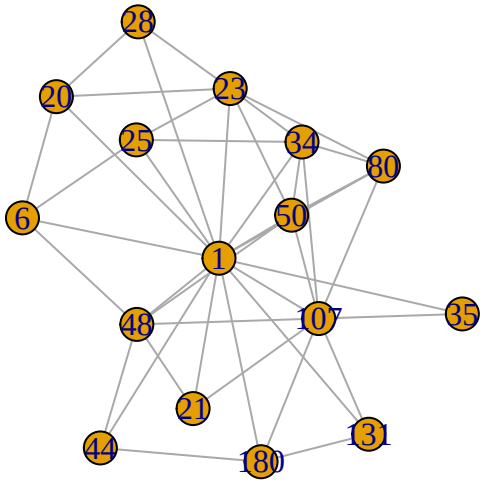
Top Ten Episodes by Similarity

The episodes which appear most similar are all sequential, a quick googling after referencing them to our data set shows that they are all multi part episodes.

Table 1: Ten Closest Pairs of Episodes

Episode1	Episode2	Similarity
256	257	0.7955508
177	178	0.8018494
164	165	0.8034759
245	246	0.8050314
244	246	0.8067941
207	208	0.8087774
142	143	0.8089330
244	245	0.8093306
206	207	0.8114169
1	97	0.8221757

```
## First Cluster:  1 6 20 21 23 25 28 34 35 44 48 50 80 107 131 180
```


28
20
23
25
34
80
6
50
1
48
107
35
21
131
44
180

Code Appendix

The code used in creating the above document:

```
##### Packages ####
# The usual suspects
library(magrittr)
library(tidyverse)

# Text analysis
library(text2vec)
library(stringr)
library(tidytext)
data("stop_words")
library(SnowballC)
library(wordcloud)

# Our very own
library(FredsVietorisRips)

# Graph adjacencies
library(igraph)

##### Import Data ####
df_original <- readr::read_csv("D:/southpark.csv")

##### Clean the set ####
# 1. Unnest tokens
# 2. Remove Stop Words
# 2. Group by season by episode
df <- df_original %>%
  unnest_tokens(word, Line) %>%
  anti_join(stop_words, by="word") %>%
  group_by(Season, Episode) %>%
  summarise(text = str_c(word, collapse = " ")) %>%
  ungroup()

# Remove weird last entry
df <- df[-258,]

##########################
# Note for the appendix #
##########################
# In the notes below,
# where we use the word "Sample"
# we mean the whole set

##### Tokenize Sample ####
it_sample <- df$text %>%
  itoken(progressbar = FALSE)

##### Vectorize (???) ####
vectorizer <- df$text %>%
  itoken(progressbar = FALSE) %>%
```

```r
  create_vocabulary() %>%
  prune_vocabulary() %>%
  vocab_vectorizer()


##### Document Term Matrix ####
dtm_sample <- create_dtm(it_sample, vectorizer)
# dim(dtm_sample)

##### Jaccard Similarity ####
jacsim <- sim2(dtm_sample, dtm_sample, method = "jaccard", norm = "none")
# dim(jacsim)

##### Convert fancy object to matrix ####
jsmat <- jacsim %>%
  as.matrix()

##### Jaccard Distance ####
# Flip it around so 1 is max distance
# and 0 indicates two sets are the same
jsmat <- 1 - jsmat
```