ConstitutionalPreambles

2024-09-16

Question 1

First, let's visualize the data to better understand how constitutional documents differ. Start by importing the preamble data, tokenizing, and preprocessing the text. Calculate both the regular document term frequency and the tf-idf weighted term frequency. In both cases, visualize the preamble to the U.S. Constitution with a word cloud. How do the results differ between the two methods?

```r
library(tidyverse)
library(tm)
library(SnowballC)
library(wordcloud)

constitutions <- read_csv("constitutions.csv")

corpus.raw <- Corpus(VectorSource(constitutions$preamble)) # load the raw corpus

corpus.prep <- tm_map(corpus.raw, content_transformer(tolower)) # make lower case
corpus.prep <- tm_map(corpus.prep, stripWhitespace) # remove whitespace
corpus.prep <- tm_map(corpus.prep, removePunctuation) # remove punctuation
corpus.prep <- tm_map(corpus.prep, removeNumbers) # remove numbers
corpus <- tm_map(corpus.prep, removeWords, stopwords("English")) # remove stop words
corpus <- tm_map(corpus, stemDocument) # stem remaining words

dtm <- DocumentTermMatrix(corpus) # Document-term matrix
dtm
```

```
## <<DocumentTermMatrix (documents: 155, terms: 3219)>>
## Non-/sparse entries: 16150/482795
## Sparsity           : 97%
## Maximal term length: 19
## Weighting          : term frequency (tf)
```

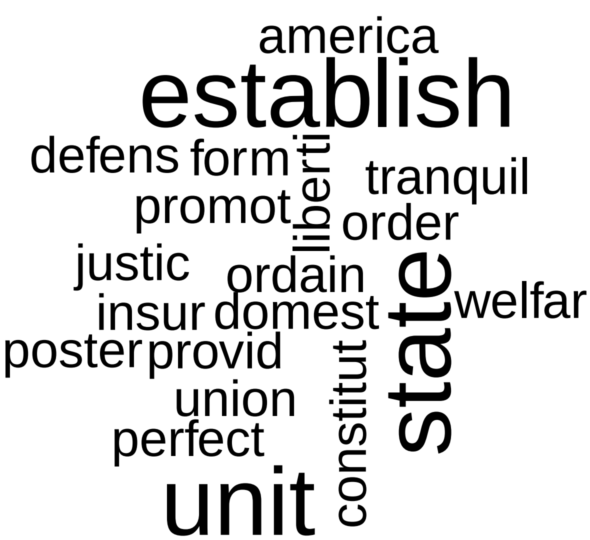
```r
dtm.tfidf <- weightTfIdf(dtm) # Calculate TF-IDF
dtm.tfidf
```

```
## <<DocumentTermMatrix (documents: 155, terms: 3219)>>
## Non-/sparse entries: 16150/482795
## Sparsity           : 97%
## Maximal term length: 19
## Weighting          : term frequency - inverse document frequency (normalized) (tf-idf)
```

```r
## Create word clouds for U.S. Constitution's preamble using the dtm and dtm.tfidf matrices.

dtm.mat <- as.matrix(dtm) # Coerce to matrix
dtm.tfidf.mat <- as.matrix(dtm.tfidf) # Coerce to matrix

## Add names to matrix
```

```r
rownames(dtm.mat) <- constitutions$country
rownames(dtm.tfidf.mat) <- constitutions$country

# word cloud for USA preamble, using tf-idf weighting (common terms across documents downweighted)
wordcloud(words = colnames(dtm.tfidf.mat),
          freq = dtm.tfidf.mat["united_states_of_america",],
          scale = c(3, 0.2),
          max.words = 20)
```


```r
# same, now weighted only by term frequency
wordcloud(words = colnames(dtm.mat),
          freq = dtm.mat[149,],
          scale = c(3, 0.2),
          max.words = 20)
```



Question 2

We next apply the k-means algorithm to the tf-idf and identify clusters of similar constitution preambles. Set the number of clusters to 4 and describe the results. To make each row comparable, divide it by a constant such that each row represents a vector of unit length. Note that the length of a vector $a = [a_1, a_2, \ldots, a_n]$ is given by $||a|| = \sqrt{a_1^2 + a_2^2 + \cdots + a_n^2}$

```r
## Function to normalize each row

normalize_row <- function(row) {
  scaled_row <- scale(row, center = FALSE)
  normalized_row <- scaled_row / sqrt(sum(scaled_row^2))
  return(normalized_row)
}

## Apply function to each row using apply()

normalized.dtm.tfidf.mat <- apply(dtm.tfidf.mat, 1, normalize_row)

## Apply the k-means algorithm with 4 clusters

kmeans.4.out <- kmeans(normalized.dtm.tfidf.mat, centers = 4, iter.max = 10, nstart = 5)

## Loop to print words and countries in each cluster

k <- 4 # number of clusters

for (i in 1:k) {
  cat("Cluster", i, "\n")
  cat("Top 10 countries:\n") # most important countries in each cluster
  print(head(sort(kmeans.4.out$centers[i, ], decreasing = TRUE), n = 10))
  cat("\n")
  cat("Top 10 words:\n") # most important words in each cluster
  print(head(sort(colnames(dtm.tfidf.mat)[kmeans.4.out$cluster == i]), n = 10))
  cat("\n")
}
```

```
## Cluster 1
## Top 10 countries:
##                             togo                            gabon
##                       0.03537998                       0.03434845
##                            niger                         djibouti
##                       0.03250592                       0.03097836
## congo_democratic_republic_of_the                          senegal
##                       0.02924602                       0.02894303
##                          burundi                     burkina_faso
##                       0.02819640                       0.02590218
##                            congo                         colombia
##                       0.02547689                       0.02374089
##
## Top 10 words:
##  [1] "access"  "accord"  "adher"   "adopt"   "affirm"  "africa"  "african"
##  [8] "amiti"   "anim"    "assembl"
##
## Cluster 2
## Top 10 countries:
##          saint_lucia            dominica              belize antigua_and_barbuda
##           0.02764541          0.02263856          0.02225110          0.02172965
## trinidad_and_tobago               kenya               ghana          seychelles
##           0.02154112          0.01799314          0.01765239          0.01748193
```

```
##             grenada    papua_new_guinea
##          0.01683810          0.01667875
##
## Top 10 words:
##  [1] "abil"       "accept"     "account"    "achiev"     "acknowledg"
##  [6] "activ"      "adequ"      "adult"      "advanc"     "aim"
##
## Cluster 3
## Top 10 countries:
## iran_islamic_republic_of                 thailand                    egypt
##              0.004958107              0.004100514              0.003736888
##                     iraq                     cuba         papua_new_guinea
##              0.003302537              0.003109424              0.002954909
##                  bahrain               cape_verde                  morocco
##              0.002797363              0.002704720              0.002670813
##                  algeria
##              0.002668931
##
## Top 10 words:
##  [1] "abandon"   "abdallah"  "abdel"     "abid"      "abl"       "abli"
##  [7] "abneg"     "abolish"   "aborigin"  "abovement"
##
## Cluster 4
## Top 10 countries:
##             tonga             nauru            serbia            tuvalu
##        0.12602886        0.10140580        0.06803568        0.05191869
##   solomon_islands            kosovo         swaziland           bahamas
##        0.04817962        0.03854845        0.02964399        0.02881294
##         sri_lanka korea_republic_of
##        0.02675950        0.02611062
##
## Top 10 words:
##  [1] "act"      "amend"    "appeal"   "court"    "day"      "eight"
##  [7] "hundr"    "island"   "kosovo"   "metohija"
```

Question 3

Apply the cosine similarity function to identify the five constitutions whose preambles most resemble that of the US constitution.

```r
## Define vector for US (x) and remove US from dtm.tfidf.mat (y)

row_index <- which(rownames(dtm.tfidf.mat) == "united_states_of_america")

if (length(row_index) == 0) {
  stop("united_states_of_america not found in row names")
}

## Define function for cosine similarity

cosine_sim <- function(a, b) {
  # Ensure a and b are matrices with at least two dimensions
  if (is.vector(a)) a <- matrix(a, nrow = 1)
  if (is.vector(b)) b <- matrix(b, nrow = 1)
```

```r
  numer <- rowSums(a * b)
  denom <- sqrt(rowSums(a^2)) * sqrt(rowSums(b^2))

  return(numer / denom)
}


## Function to compute similarity

compute_similarity <- function(matrix, index) {
  row_vector <- matrix[index, ]
  matrix_removed <- matrix[-index, ]
  country_names <- colnames(matrix_removed)
  broadcasted_vector <- matrix(rep(row_vector, each = nrow(matrix_removed)),
                               nrow = nrow(matrix_removed),
                               byrow = TRUE)
  similarities <- cosine_sim(broadcasted_vector, matrix_removed)
  named_similarities <- setNames(similarities, country_names)
  return(named_similarities)
}

## Compute similarity
similarities <- compute_similarity(normalized.dtm.tfidf.mat, row_index)

# Display top 5 similar countries
top_5 <- head(sort(similarities, decreasing = TRUE), n = 5)
top_5_names <- names(top_5)
top_5_names <- str_to_title(top_5_names)
print(top_5)
```

```
##  bulgaria     gabon    gambia   eritrea     egypt
## 0.8441370 0.6612370 0.6250555 0.5920004 0.5918480
```

```r
paste("The five countries with preambles most similar to the preamble of the US Constitution are:",
                paste(top_5_names, collapse = ", "))
```

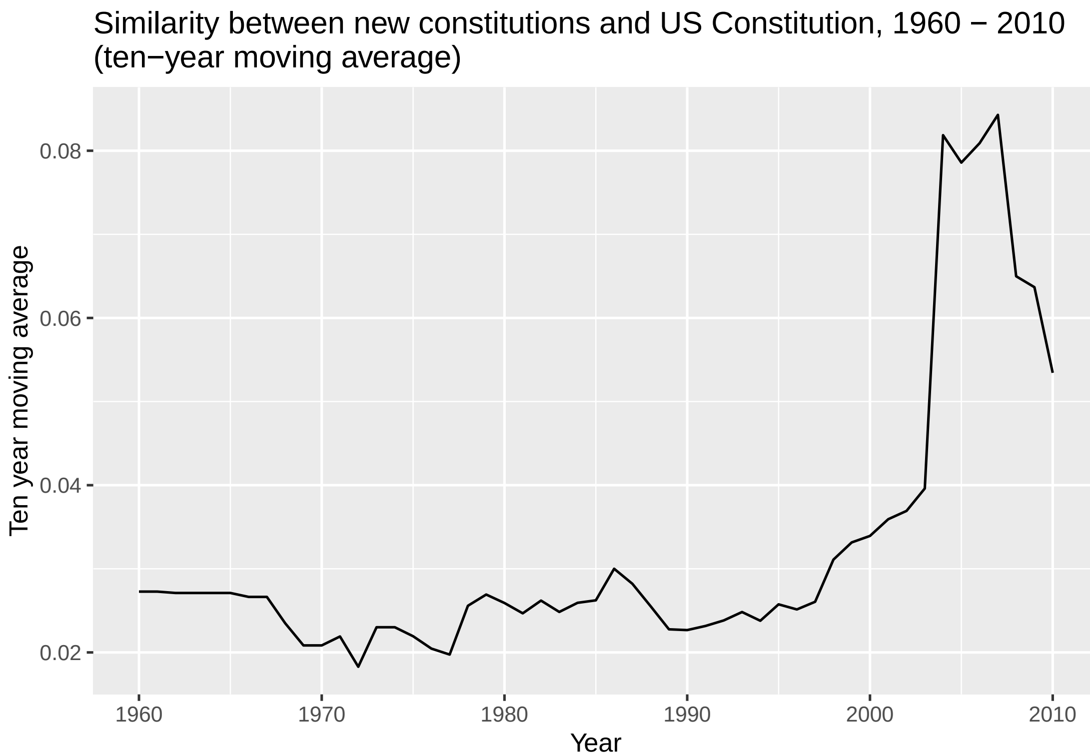
```
## [1] "The five countries with preambles most similar to the preamble of the US Constitution are: Bulga
```

Question 4

We examine the influence of US constitutions on other constitutions over time. We focus on the post-war period. Sort the constitutions chronologically and calculate, for every ten years from 1960 until 2010, the average of cosine similarity between the US constitution and the constitutions that were created during the past decade. Plot the result. Each of these averages computed over time is called the *moving average*. Does similarity tend to increase, decrease, or remain the same over time? Comment on the pattern you observe.

```r
## Define convert-to-tfidf function

convert_2_tfidf <- function(df) {
  corpus.raw <- Corpus(VectorSource(df$preamble)) # load the raw corpus
  corpus.prep <- tm_map(corpus.raw, content_transformer(tolower)) # make lower case
  corpus.prep <- tm_map(corpus.prep, stripWhitespace) # remove whitespace
  corpus.prep <- tm_map(corpus.prep, removePunctuation) # remove punctuation
  corpus.prep <- tm_map(corpus.prep, removeNumbers) # remove numbers
  corpus <- tm_map(corpus.prep, removeWords, stopwords("English")) # remove stop words
```

```r
  corpus <- tm_map(corpus, stemDocument) # stem remaining words
  dtm <- DocumentTermMatrix(corpus) # Document-term matrix
  dtm.tfidf <- weightTfIdf(dtm) # Calculate TF-IDF
  dtm.tfidf.mat <- as.matrix(dtm.tfidf) # coerce to matrix
  return(dtm.tfidf.mat)
}

## Define moving average function

tenyr_moving_avg_cosine_sim <- function(df, name, start_year, end_year, window_size) {
  dtm.tfidf.mat <- convert_2_tfidf(df) # Convert to tfidf
  rownames(dtm.tfidf.mat) <- df$country # Add names to matrix
  ordered <- dtm.tfidf.mat[order(df$year),] # Order by year
  row_index <- ordered[name, , drop = FALSE] # Keep as matrix

  ma <- rep(NA, end_year - start_year + 1) # initialize moving average

  for (i in start_year:end_year) {
    start <- i - window_size + 1
    end <- i
    window <- ordered[df$year >= start & df$year <= end, , drop = FALSE]

    # Ensure row_index is compared to each row in the window
    cosine_sims <- apply(window, 1, function(row) cosine_sim(row_index, row))

    ma[i - start_year + 1] <- mean(cosine_sims)
  }

  return(ma)
}

ma <- tenyr_moving_avg_cosine_sim(constitutions, "united_states_of_america", 1960, 2010, 10)
ma <- tibble(year = 1960:2010, moving_avg = ma)

plot_ma_1960_2010 <- ggplot(ma, aes(year, moving_avg)) +
  geom_line() +
  labs(x = "Year",
       y = "Ten year moving average",
       title = "Similarity between new constitutions and US Constitution, 1960 - 2010 \n(ten-year moving
plot_ma_1960_2010
```



average-1.pdf

We see that the trend is roughly flat until around 1995, while the similarity increases to around 0.08 around the year 2005.

Question 5

We next construct directed, weighted network data based on the cosine similarity of constitutions. Specifically, create an adjacency matrix whose (i,j)-th entry represents the cosine similarity between the i-th and j-th constitution preambles, where the i-th constitution was created in the same year or after the j-th constitution. This entry equals zero if the i-th constitution was created before the j-th constitution. After creating the graph object, you can assign weights to the edges where the weight is the cosine similarity. Apply the PageRank algorithm to this weighted adjacency matrix.

```
## Create adjacency matrix

library("igraph")
n <- nrow(constitutions)
similarity.cons.adj <- matrix(0, nrow = n, ncol = n)
colnames(similarity.cons.adj) <- rownames(similarity.cons.adj) <- constitutions$country

## Define cosine_sim_v for two vectors

cosine_sim_v <- function(a,b){
  require(pracma)
  numer <- dot(a,b)
  denom <- Norm(a) * Norm(b)
  return(numer / denom)
}

## Create adjacency matrix for constitutional similarities w/ nested for loops

for (i in 1:n){
```

```r
  for (j in 1:n) {
    if (constitutions$year[i] >= constitutions$year[j]) {
    similarity <- cosine_sim_v(dtm.tfidf.mat[i,], dtm.tfidf.mat[j,])
    similarity.cons.adj[i, j] <- similarity
    } else {
    similarity.cons.adj[i, j] <- 0
    }
  }
}

similarity.cons.adj.graph <- graph_from_adjacency_matrix(similarity.cons.adj, mode = "directed", weighte

constitutions$indegree <- degree(similarity.cons.adj.graph, mode = "in")
constitutions$outdegree <- degree(similarity.cons.adj.graph, mode = "out")

pr_constitutions <- page_rank(similarity.cons.adj.graph, directed = TRUE)
constitutions$pr <- pr_constitutions$vector

# Most influential constitutions according to PageRank algorithm

most_influential <- constitutions %>%
  select(country, pr) %>%
  arrange(desc(pr)) %>%
  slice_head(n = 10)
most_influential
```

```
## # A tibble: 10 x 2
##    country                       pr
##    <chr>                      <dbl>
##  1 united_states_of_america 0.117
##  2 argentina                0.0636
##  3 latvia                   0.0335
##  4 tonga                    0.0331
##  5 ireland                  0.0295
##  6 japan                    0.0266
##  7 indonesia                0.0216
##  8 india                    0.0212
##  9 taiwan                   0.0203
## 10 korea_republic_of        0.0197
```

```r
# Least influential constitutions according to PageRank algorithm

least_influential <- constitutions %>%
  select(country, pr) %>%
  arrange(desc(pr)) %>%
  slice_tail(n = 10)
least_influential
```

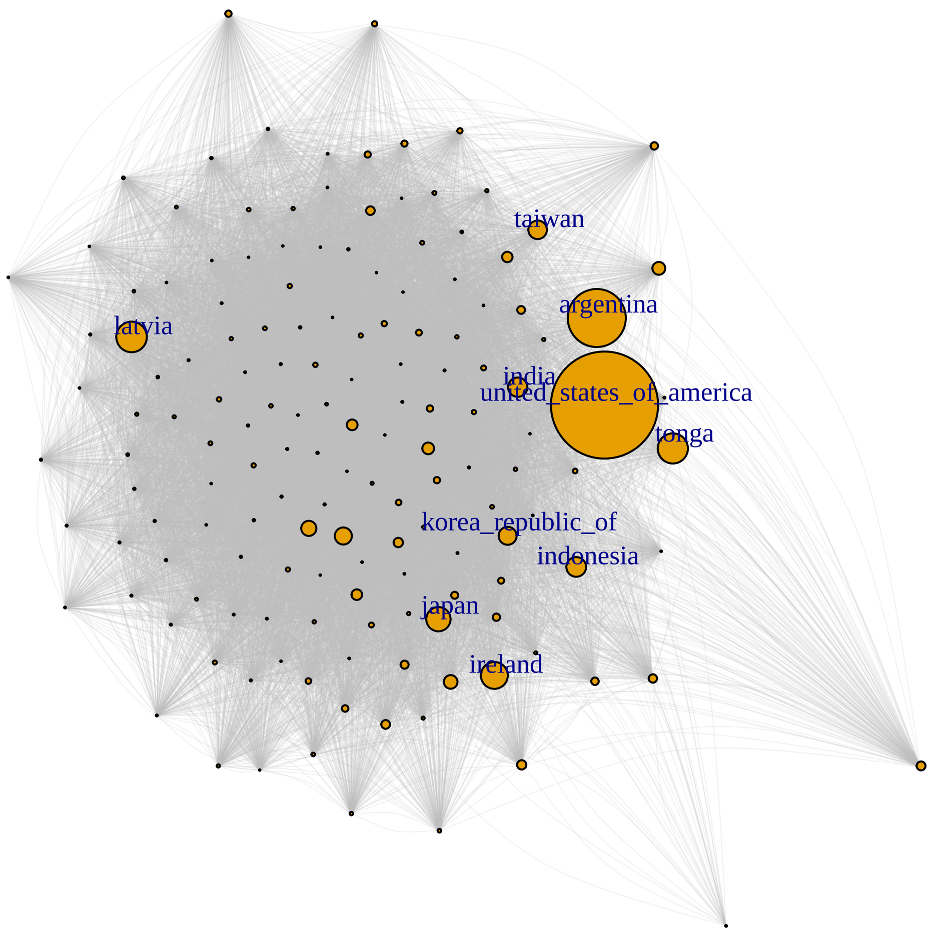
```
## # A tibble: 10 x 2
##    country                        pr
##    <chr>                       <dbl>
##  1 morocco                   0.00174
##  2 south_sudan               0.00171
##  3 bhutan                    0.00170
##  4 syrian_arab_republic      0.00170
```

```
##   5 zimbabwe                 0.00167
##   6 tunisia                  0.00165
##   7 central_african_republic 0.00165
##   8 egypt                    0.00164
##   9 fiji                     0.00164
##  10 thailand                 0.00163
```

We see that the ten most influential Constitutions have been those of the United States, Argentina, Latvia, Tonga, Ireland, Japan, Indonesia, India, Taiwan, and the Republic of Korea. The ten least influential constitutions have been Morocco, South Sudan, Bhutan, Syria, Zimbabwe, Tunisia, the Central African Republic, Fiji, and Thailand.

```r
col <- adjustcolor("grey", alpha.f = 0.2)
top_10_indices <- order(constitutions$pr, decreasing = TRUE)[1:10]
vertex_labels <- rep(NA, length(constitutions$pr))
vertex_labels[top_10_indices] <- constitutions$country[top_10_indices]

plot(similarity.cons.adj.graph,
     layout = layout_with_fr,
     vertex.size = constitutions$pr * 200,
     vertex.label = vertex_labels,
     vertex.label.cex = 0.8,
     vertex.label.dist = 0.5,
     edge.arrow.size = 0.01,
     edge.color = col,
     edge.width = 0.5,
     edge.curved = 0.3,
     margin = 0.2,
     main = "Graph of Constitutional Influence")
```

Graph of Constitutional Influence



page rank-1.pdf