Text As Data HW 1

Jean An (cya220)

2/21/2022

```
library(tidyverse)
library(quanteda)
library(quanteda.corpora)
library(quanteda.textstats)
library(quanteda.textplots)
```

Q1

```
speeches <- corpus_subset(data_corpus_inaugural, President == "Reagan")
tokenized <- tokens(speeches, remove_punct = TRUE)
```

(a)

```
TTR_calculator <- function(tokens_item) {
  corpused <- sapply(tokens_item, paste, collapse=" ") %>% corpus()
  corpus_info <- summary(corpused)
  corpus_info %>%
    mutate(TTR = Types / Tokens) %>%
    select(Text, TTR)
}
TTR_calculator(tokenized)
```

```
##          Text       TTR
## 1 1981-Reagan 0.3680099
## 2 1985-Reagan 0.3568643
```

(b)

```
reagan_dfm <- dfm(tokenized, tolower = FALSE)
textstat_simil(reagan_dfm, margin = "documents", method = "cosine")
```

```
## textstat_simil object; method = "cosine"
##             1981-Reagan 1985-Reagan
## 1981-Reagan       1.000       0.956
## 1985-Reagan       0.956       1.000
```

The cosine similarity between the two documents is 0.956.

Q2

(a) I believe the TTR will decrease a little bit, but not much. This is because words with the same stems will now be grouped into the same types, while the number of tokens remains unchanged. The similarity between the two documents should also be relatively unaffected, because stemming the words simply groups more tokens together, and doesn't really create new words.

```
stemmed_token <- tokens_wordstem(tokenized)
TTR_calculator(stemmed_token)
```

```
##          Text       TTR
## 1 1981-Reagan 0.3322368
## 2 1985-Reagan 0.3178627
```

```
stemmed_dfm <- dfm(stemmed_token, tolower = FALSE)
textstat_simil(stemmed_dfm, margin = "documents", method = "cosine")
```

```
## textstat_simil object; method = "cosine"
##             1981-Reagan 1985-Reagan
## 1981-Reagan       1.000       0.957
## 1985-Reagan       0.957       1.000
```

(b) I believe the TTR would increase drastically. This is because by removing the stop words, we are removing a large number of tokens (assuming Reagan speaks with lots of stop words like a normal person would) that are grouped into a few types. The similarity between the two documents should greatly decrease, because he likely used similar stop words in both of his speeches, and by removing these a major part of the similarities are discarded.

```
nostop_token <- tokens_remove(tokenized, pattern = stopwords("en"))
TTR_calculator(nostop_token)
```

```
##          Text       TTR
## 1 1981-Reagan 0.6608544
## 2 1985-Reagan 0.6059908
```

```
nostop_dfm <- dfm(nostop_token, tolower = FALSE)
textstat_simil(nostop_dfm, margin = "documents", method = "cosine")
```

```
## textstat_simil object; method = "cosine"
##             1981-Reagan 1985-Reagan
## 1981-Reagan       1.000       0.668
## 1985-Reagan       0.668       1.000
```

(c) I believe the TTR would decrease, but only very little. By converting all words to lowercase, the number of tokens remain unchanged, and the only words that are affected and grouped into the same type are likely just words that happen to be the start of sentences and certain proper nouns. The similarity between the two documents would also be largely unaffected, because the words are basically still the same, and in rare occasion a word after being lowercased becomes another word.

```r
lower_token <- tokens_tolower(tokenized)
TTR_calculator(lower_token)
```

```
##          Text       TTR
## 1 1981-Reagan 0.3466283
## 2 1985-Reagan 0.3377535
```

```r
lower_dfm <- dfm(lower_token, tolower = TRUE)
textstat_simil(lower_dfm, margin = "documents", method = "cosine")
```

```
## textstat_simil object; method = "cosine"
##             1981-Reagan 1985-Reagan
## 1981-Reagan       1.000       0.959
## 1985-Reagan       0.959       1.000
```

(d) I think tf-idf makes some sense, but might not be as useful given that our corpus only has two documents. This means that every word Reagan used in both speeches would have a weight of zero, because idf = 0; while every word he used in one speech but not another has idf = 0.69. Therefore, the tf-idf is essentially solely dependent on the tf, and not the idf.

```r
topfeatures(dfm_tfidf(reagan_dfm))
```

```
##   weapons   nuclear ourselves    reduce        To    beyond     means     price
##   1.80618   1.80618   1.50515   1.50515   1.20412   1.20412   1.20412   1.20412
##   special     these
##   1.20412   1.20412
```

```r
topfeatures(dfm_tfidf(stemmed_dfm))
```

```
##  nuclear children     mean  ourselv     turn       To maintain   beyond
##  1.80618  1.50515  1.50515  1.50515  1.50515  1.20412  1.20412  1.20412
##    price  special
##  1.20412  1.20412
```

```r
topfeatures(dfm_tfidf(nostop_dfm))
```

```
##  weapons  nuclear   reduce   beyond    means    price  special      ago
##  1.80618  1.80618  1.50515  1.20412  1.20412  1.20412  1.20412  1.20412
##   better increase
##  1.20412  1.20412
```

```r
topfeatures(dfm_tfidf(lower_dfm))
```

```
##   weapons   nuclear ourselves     union    reduce     means     price   special
##   1.80618   1.80618   1.50515   1.50515   1.50515   1.20412   1.20412   1.20412
##       ago    better
##   1.20412   1.20412
```

We can see that the terms "nuclear" and "weapons" have the highest weights, which suggest Reagan likely mentioned nuclear weapons many times in one speech and not at all in the other. Interestingly, when stemmed, "weapons" is no longer one of the top features. My guess is that he used "weapons" many times in one speech and mentioned "weapon" (but not "weapons") in the other speech, and once "weapons" is stemmed it becomes the same as "weapon" and received a weight of zero.

Q3

```r
headlines <- c(headline1 = "Nasa Mars rover: Perseverance robot all set for big test.",
               headline2 = "NASA Lands Its Perseverance Rover on Mars.")
head_token <- tokens(headlines, remove_punct = TRUE)
head_dfm <- dfm(head_token, tolower = TRUE)
```

In the preprocessing, I chose to remove punctuation and convert all words to lower case. This is because punctuation and capitalization in headlines often times simply reflects the style of editing, rather than provide any useful information.

(a) Euclidean Distance

```r
sqrt(sum((head_dfm[1,] - head_dfm[2,])^2))
```

```
## [1] 3
```

(b) Manhattan Distance

```r
sum(abs(head_dfm[1,] - head_dfm[2,]))
```

```
## [1] 9
```

(c) Cosine Similarity

```r
product <- sum(head_dfm[1,] * head_dfm[2,])
norm_1 <- sqrt(sum(head_dfm[1,]^2))
norm_2 <- sqrt(sum(head_dfm[2,]^2))
product / (norm_1 * norm_2)
```

```
## [1] 0.4780914
```

(d) Levenshtein Distance

The Levenshtein distance is 3. To go from *robot* to *rover*, we have to replace *b* with *r*, replace *o* with *e*, and replace *t* with *r*. That is a total of three operations.

Q4

```
library(gutenbergr)
library(stylest)
set.seed(1984L)
```

(a)

```
author_list <- c("Poe, Edgar Allan", "Twain, Mark",
                 "Shelley, Mary Wollstonecraft","Doyle, Arthur Conan")
book_list <- c(932,1064,1065,32037,74,76,86,91,84,6447,15238,18247,108,126,139,244)

prepare_dt <- function(book_list, num_lines, removePunct = TRUE){
     meta <- gutenberg_works(gutenberg_id == book_list)
     meta <- meta %>% mutate(author = unlist(str_split(author, ","))[1]
     %>% tolower(.))
     texts <- lapply(book_list, function(x) gutenberg_download(x,
     mirror="http://mirrors.xmission.com/gutenberg/") %>%
                       #select(text) %>%
                       sample_n(500, replace=TRUE) %>%
                       unlist() %>%
                       paste(., collapse = " ") %>%
                       str_replace_all(., "^ +| +$|( ) +", "\\1"))
     # remove apostrophes
     texts <- lapply(texts, function(x) gsub("‘|’", "", x))
     if(removePunct) texts <- lapply(texts, function(x)
     gsub("[^[:alpha:]]", " ", x))
     # remove all non-alpha characters
     output <- tibble(title = meta$title, author = meta$author, text =
     unlist(texts, recursive = FALSE))
}
texts_dt <- lapply(book_list, prepare_dt, num_lines = 500, removePunct = TRUE)
```

(b)

```
texts_dt <- do.call(rbind, texts_dt)
str(texts_dt)
```

```
## tibble [16 x 3] (S3: tbl_df/tbl/data.frame)
##  $ title : chr [1:16] "The Fall of the House of Usher" "The Masque of the Red Death" "The Raven" "Eu
##  $ author: chr [1:16] "poe" "poe" "poe" "poe" ...
##  $ text  : chr [1:16] "
```

(c)

```r
stopwords_en <- stopwords("en")
filter <- corpus::text_filter(drop_punct = TRUE, drop_number = TRUE, drop = stopwords_en)
vocab_terms <- stylest_select_vocab(texts_dt$text, texts_dt$author,
                                    filter = filter, smooth = 1, nfold = 5,
                                    cutoff_pcts = c(25, 50, 75, 99))
vocab_terms$cutoff_pct_best
```

```
## [1] 75
```

```r
vocab_terms$miss_pct
```

```
##          [,1]     [,2]     [,3]     [,4]
## [1,] 33.33333 33.33333 33.33333 33.33333
## [2,] 33.33333 33.33333  0.00000 33.33333
## [3,] 25.00000 25.00000 25.00000 25.00000
## [4,] 50.00000 50.00000 50.00000  0.00000
## [5,] 25.00000 25.00000 25.00000 50.00000
```

(d)

```r
vocab_subset <- stylest_terms(texts_dt$text, texts_dt$author,
                              vocab_terms$cutoff_pct_best , filter = filter)
style_model <- stylest_fit(texts_dt$text, texts_dt$author,
                           terms = vocab_subset, filter = filter)
authors <- unique(texts_dt$author)
term_usage <- style_model$rate
lapply(authors, function(x) head(term_usage[x,][order(-term_usage[x,])], 5)) %>%
  setNames(authors)
```

```
## $poe
##        upon        door         one     chamber         now
## 0.010938874 0.006981091 0.006871152 0.006761214 0.006431398
##
## $twain
##           t           s         tom         got        said
## 0.024639678 0.013830013 0.008213226 0.008001272 0.007895295
##
## $shelley
##         one           s         now         may        love
## 0.006079845 0.005326590 0.004788551 0.004465727 0.004142903
##
## $doyle
##        said        upon         one           s          us
## 0.010603680 0.010267056 0.008920557 0.006564183 0.005778725
```

It's hard for me to judge if these terms make sense or not, because I haven't read many of these selected books. However, I know that having the term "tom" in the top-5 for Mark Twain when we included the book *The Adventures of Tom Sawyer* and *Tom Sawyer Abroad* certainly makes sense.

(e)

```
test <- data.frame(term_usage) %>%
  filter(rownames(term_usage) %in% c('twain','doyle'))
rate_ratio <- term_usage['twain',] / term_usage['doyle',]
head(rate_ratio[order(-rate_ratio)], 5)
```

```
##     says      jim      ain      tom       en
## 74.61297 72.72404 59.50148 48.79751 42.50106
```

I would interpret this ordering as “Mark Twain used the term ‘says’ 74 times more than Arthur Conan Doyle; this is the largest ratio for Twain over Doyle.” and so on. We also see the terms ‘jim’ and ‘tom’, which make sense, because Jim is one of the main characters in *Adventures of Huckleberry Finn*, while Tom is the main character in the two Tom Sawyer books mentioned before.

(f)

```
mys_file <- "mystery_excerpt.rds"
mystery_excerpt <- readRDS(mys_file)

pred <- stylest_predict(style_model, mystery_excerpt)
pred$predicted
```

```
## [1] twain
## Levels: doyle poe shelley twain
```

Based on the prediction of the model, Mark Twain is the most likely author of this excerpt.

(g)

```
texts_tokens <- tokens(texts_dt$text) %>% setNames(texts_dt$title)

texts_lambda <- textstat_collocations(texts_tokens, min_count = 5) %>%
  arrange(desc(lambda))
head(texts_lambda, 10)
```

```
##              collocation count count_nested length   lambda        z
## 667          edgar allan     7            0      2 14.64330 7.202582
## 685      denser perfumed     6            0      2 14.50022 7.114569
## 686      whispering vows     6            0      2 14.50022 7.114569
## 709 syllable expressing     5            0      2 14.33318 7.009050
## 548     candelabrum amid     6            0      2 13.40159 7.979787
## 549        unseen censer     6            0      2 13.40159 7.979787
## 524            allan poe     7            0      2 13.03384 8.188891
## 557    arabesque figures     5            0      2 12.72371 7.918627
## 558      densely crowded     5            0      2 12.72371 7.918627
## 559       unsuited limbs     5            0      2 12.72371 7.918627
```

```
texts_count <- textstat_collocations(texts_tokens, min_count = 5) %>%
  arrange(desc(count))
head(texts_count, 10)
```

```
##      collocation count count_nested length    lambda         z
## 1         of the   692            0      2 1.9096885 40.162016
## 6         in the   320            0      2 1.7353212 26.037857
## 1093     and the   244            0      2 0.3679152  5.362239
## 175       to the   228            0      2 0.8517698 11.743742
## 2         it was   208            0      2 3.1108918 36.624131
## 18        on the   130            0      2 2.1815977 19.780368
## 408         of a   120            0      2 0.8819775  9.099416
## 26      from the   115            0      2 1.9740319 17.432193
## 8          to be   112            0      2 3.0204174 25.893505
## 1155    that the   101            0      2 0.5357825  5.071165
```

I don't think any of these sets of bi-grams are multi-word expressions. Most of the ones with a high lambda value are pairs of adverb-verbs or adjective-nouns. All of the ones with a high count are just conjunctions.

Q5

```
library("sophistication")
data(data_corpus_ungd2017, package = "quanteda.corpora")
```

(a)

```
snippetData <- snippets_make(data_corpus_ungd2017, nsentence = 1, minchar = 150, maxchar = 350)
snippetData <- snippets_clean(snippetData)
head(snippetData, 10)
```

```
##          docID snippetID
## 1  Afghanistan    100001
## 2  Afghanistan    100002
## 3  Afghanistan    100003
## 4  Afghanistan    100009
## 5  Afghanistan    100011
## 6  Afghanistan    100012
## 7  Afghanistan    100015
## 8  Afghanistan    100016
## 9  Afghanistan    100017
## 10 Afghanistan    100020
##
## 1                                                       As I stand here before the General Assembly to
## 2                                  Shaped by the Great Depression and tempered by the carnage of the Secon
## 3                      The United Nations, the International Monetary Fund, the World Bank and other orga
## 4                                                              There is an emerging consensus that adv
## 5                                                               Sixteen years after the tragedy of 11
## 6      Driven by transnational terrorist networks, criminal organizations, cybercrime and State spons
## 7  Terrorism is not only an attack on human life and basic freedoms, but an attack on the compact of
## 8                                                                        We must confront the threat o
## 9                               Lastly, despite the incorporation of tenets of the Universal Declaration o
## 10                                                                         I welcome the chance for Af
```

(b)

```
testData <- sample_n(snippetData, 1000)
snippetPairsMST <- pairs_regular_make(testData)
gold_questions <- pairs_gold_make(snippetPairsMST, n.pairs = 10)

print(gold_questions$text1)
```

```
##  [1] "Peacekeeping reform in particular requires a carefully tailored approach, without abrupt shift
##  [2] "If that endeavour is successful, we will be honoured to work even harder for the advancement o
##  [3] "Bulgaria categorically condemns the repeated nuclear tests and missile launches by the Democra
##  [4] "We contemplate how we can best find grand solutions, when all we really need is to translate t
##  [5] "Ghana will also continue to be active in the multilateral organizations to which we belong, su
##  [6] "For those who question the veracity of that science, the cluster of extreme weather events ove
##  [7] "Once again, Cameroon, as it did from this very rostrum on 10 September 2000, urges the world t
##  [8] "Eliminating radicalism and religious fundamentalism should also be a major priority for our St
##  [9] "Practical approaches could allow us to work through existing controversies in order to achieve
## [10] "On behalf of the people and the Government of the Republic of Paraguay, I wish to express to t
```

```
print(gold_questions$text2)
```

```
##  [1] "Others are economic migrants prepared to risk everything on perilous sea crossings in the desp
##  [2] "Above and beyond those urgent humanitarian actions, Monaco's cooperation system implements a p
##  [3] "Let me end by reciting a verse that is a synthesis of our thought: \"May all be happy; may all
##  [4] "Accordingly, this year has witnessed numerous initiatives for fruitful cooperation, notably th
##  [5] "We all want to be a part of the EU, but sometimes people in the Balkans and people in Serbia a
##  [6] "The Holy See therefore appreciates the Secretary-General's explicit and strong emphasis on pre
##  [7] "Protracted conflicts require a holistic United Nations response, encompassing preventive diplo
##  [8] "No country has the right to make the world an unsafe place to live in, and we owe it to our pe
##  [9] "We are also committed to supporting efforts to make the Council more transparent and promote t
## [10] "We have disbursed $500 million for the Syria crisis since 2016, which means that we are on tra
```

My selections (in order): 2 1 2 1 2 2 1 2 2 2

```
print(gold_questions$easier_gold)
```

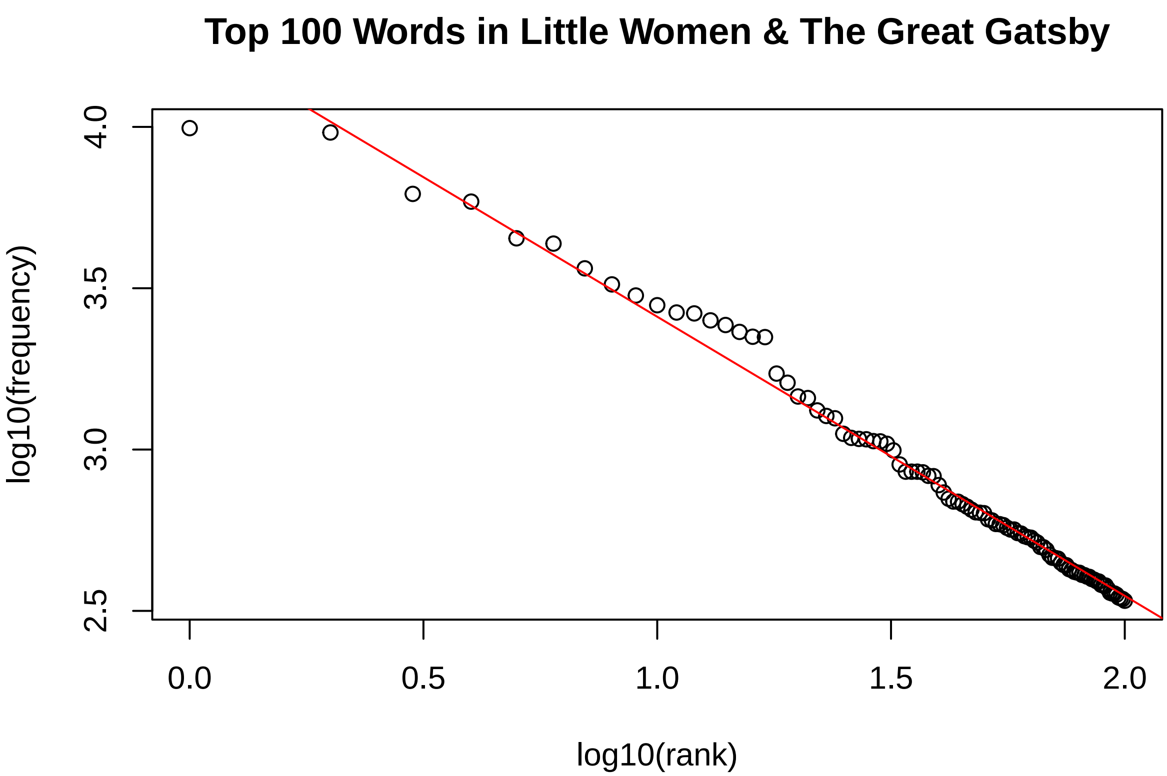
```
##  [1] 2 1 2 1 2 1 1 2 2 2
```

I was in agreement with the automated classification in nine of the ten gold pairs; the lone disagreement occurred on pair number 6. The difference in judgement likely comes from the fact that text 2 in pair 6 used some proper nouns and was also a longer text, so the classification treated it as harder to read, but I felt like some vocabularies in text 1 in pair 6 was slightly harder to interpret its true meaning.

Q6

```
little_women <- tokens(corpus(gutenberg_download(514)), remove_punct = TRUE)
little_dfm <- dfm(little_women)
great_gatsby <- tokens(corpus(gutenberg_download(64317)), remove_punct = TRUE)
great_dfm <- dfm(great_gatsby)
combined_dfm <- rbind(little_dfm, great_dfm)

plot(log10(1:100), log10(topfeatures(combined_dfm, 100)),
     xlab = "log10(rank)", ylab = "log10(frequency)",
     main = "Top 100 Words in Little Women & The Great Gatsby")
regression <- lm(log10(topfeatures(combined_dfm, 100)) ~ log10(1:100))
abline(regression, col = "red")
```



The only preprocessing decision I made is to remove punctuation. I chose not to remove stopwords because I believe stopwords have a smaller impact on texts such as novels, compared to texts such as speeches. Many of the unnecessary stopwords are likely removed in the editing process.

Q7

```r
num_tokens <- sum(lengths(little_women), lengths(great_gatsby))
M <- nfeat(combined_dfm)
k <- 44
b <- 0.4645909
k * (num_tokens)^b
```

```
## [1] 13758
```

```r
M
```

```
## [1] 13758
```

```r
print(b)
```

```
## [1] 0.4645909
```

No additional preprocessing were done for this question.

Q8

```r
head(kwic(little_women, pattern = "poor*"))
```

```
## Keyword-in-context with 6 matches.
##    [text74, 6]   It's so dreadful to be | poor | sighed Meg looking down at
##   [text188, 1]                          | Poor | Jo It's too bad but
##   [text575, 5] Goodness only knows Some | poor | creeter came a-beggin and your
##   [text639, 8]     away from here lies a | poor | woman with a little newborn
##  [text649, 11]  carry the things to the | poor | little children asked
##   [text666, 2]                         A | poor | bare miserable room it was
```

```r
head(kwic(great_gatsby, pattern = "poor*"))
```

```
## Keyword-in-context with 6 matches.
##    [text700, 6]  It's a libel I'm too | poor |
##   [text1848, 5]       Well if you're a | poor | driver you oughtn't to try
##   [text1947, 8] and felt it in others | poor | young clerks who
##   [text3277, 1]                        | poor | get children In the meantime
##  [text4315, 13]  he had just got some | poor |
##   [text4574, 4]          because I was | poor | and she was tired of
```

```r
head(kwic(little_women, pattern = "rich*", window = 4))
```

```
## Keyword-in-context with 6 matches.
##    [text132, 5]          father if he isn't | rich | and insult you when
##   [text724, 12]         the theater and not | rich | enough to
##   [text875, 10]           because he is not | rich | They shout and
##  [text1583, 10] nursery governess and felt | rich | with her
##   [text1597, 2]                         how | rich | she was in the
##   [text1603, 9]    being remembered in the | rich | old lady's will but
```

```r
head(kwic(great_gatsby, pattern = "rich*", window = 4))
```

```
## Keyword-in-context with 6 matches.
##    [text192, 7]    played polo and were | rich | together This was a
##    [text713, 1]                         | rich | nevertheless I was confused
##  [text2131, 11]             it It was a | rich | cream colour bright
##   [text2630, 2]                     and | rich | and wild but she
##   [text3162, 1]                         | rich | heap mounted higher shirts
##   [text3276, 8] and nothing's surer The | rich | get richer and the
```

```r
head(kwic(little_women, pattern = "wealth*", window = 4))
```

```
## Keyword-in-context with 6 matches.
##    [text879, 8]  that she bequeaths untold | wealth | to the young pair
##  [text4390, 10] ancient name and boundless | wealth | in return for
##   [text7983, 2]                     talent | wealth | or beauty And Amy
##  [text11549, 1]                            | Wealth | is certainly a most
##  [text12663, 4]                or women of | wealth | and position we might
##  [text18993, 3]                 the better | wealth | of love confidence and
```

```r
head(kwic(great_gatsby, pattern = "wealth*", window = 3))
```

```
## Keyword-in-context with 4 matches.
##    [text183, 6] family were enormously | wealthy | even in college
##    [text188, 1]                        | wealthy | enough to do
##  [text2163, 11]            son of some | wealthy | people in the
##   [text5272, 3]           mystery that | wealth  | imprisons and preserves
```

```r
head(kwic(little_women, pattern = "money*", window = 4))
```

```
## Keyword-in-context with 6 matches.
##    [text93, 5]     ought not to spend | money | for pleasure when our
##   [text110, 7] say anything about our | money | and she won't wish
##  [text141, 13]        wish we had the | money |
##   [text147, 4]         spite of their | money |
##   [text618, 4]            gave all my | money | to get it and
##   [text881, 8]  several quarts of tin | money | shower down upon the
```

```r
head(kwic(great_gatsby, pattern = "money*", window = 4))
```

```
## Keyword-in-context with 6 matches.
##   [text132, 11]          and gold like new | money | from the mint
##    [text184, 3]               freedom with | money | was a matter for
##    [text927, 9] Tom decisively Here's your | money | Go and buy
##  [text1017, 13]           they think of is | money | I
##   [text1086, 4]              where all his | money | comes from
##   [text1382, 2]                       easy | money | in the vicinity and
```

Based on the Key Words in Context chosen, it seems like *Little Women* discusses the concept of poverty in a more positive way, while *The Great Gatsby* promotes the ideas of being rich and wealthy. It seems like perhaps the two books regard the topic of class from different perspectives, and therefore have different conclusions.

Q9

```
data("data_corpus_ukmanifestos")
manifestos <- corpus_subset(data_corpus_ukmanifestos, Party == "Con")
sent_tokens <- unlist(tokens(manifestos, what = "sentence", include_docvars = TRUE))
yearnames <- list(unlist(names(sent_tokens)))
yearnames <- lapply(yearnames[[1]], function(x){strsplit(x, "_")[[1]][3]})
yearslist <- unlist(yearnames)
years <- unique(yearslist)
sentences_df <- tibble(text = sent_tokens, year = yearslist)
sentences_df <- sentences_df[grepl( ("[\\.\\?\\!]$"), sentences_df$text), ]
sent_corp <- corpus(sentences_df$text)
docvars(sent_corp, field = "Year") <- sentences_df$year
```

(a)

```
library(pbapply)
iters <- 10
boot_flesch <- function(grouping){
  N <- nrow(grouping)
  bootstrap_sample <- corpus_sample(corpus(c(grouping$text)), size = N, replace = TRUE)
  bootstrap_sample<- as.data.frame(as.matrix(bootstrap_sample))
  readability_results <- textstat_readability(bootstrap_sample$V1, measure = "Flesch")
  return(mean(readability_results$Flesch))
}
boot_flesch_year <- pblapply(years, function(x){
  sub_data <- sentences_df %>% filter(year == x)
  output_flesch <- lapply(1:iters, function(i) boot_flesch(sub_data))
  return(unlist(output_flesch))})
year_means <- lapply(boot_flesch_year, mean) %>% unname() %>% unlist()
year_ses <- lapply(boot_flesch_year, sd) %>% unname() %>% unlist()
estimates <- data.frame(year = years,
                        mean = round(year_means, 2),
                        ses = round(year_ses, 2))
estimates
```

```
##    year  mean  ses
## 1  1945 49.22 1.61
## 2  1950 43.76 1.19
## 3  1951 52.36 2.28
## 4  1955 49.15 1.17
## 5  1959 49.33 0.98
## 6  1964 45.80 1.67
## 7  1966 46.06 1.47
## 8  1970 45.72 1.12
## 9  1974 42.12 0.43
## 10 1979 47.44 0.40
## 11 1983 47.23 0.85
## 12 1987 46.91 0.62
## 13 1992 46.10 0.60
## 14 1997 50.10 0.69
## 15 2001 48.91 0.94
## 16 2005 49.57 1.17
```

(b)

```
flesch_score <- sentences_df$text %>%
  textstat_readability(measure = "Flesch") %>%
  group_by(sentences_df$year) %>%
  summarise(mean_flesch = round(mean(Flesch), 2)) %>%
  setNames(c("year", "mean")) %>% arrange(year)

estimates %>% select(-ses) %>%
  left_join(flesch_score, by = "year",
            suffix = c(".boot",".noboot")) %>%
  mutate(diff = mean.boot - mean.noboot)
```

```
##    year mean.boot mean.noboot  diff
## 1  1945     49.22       48.97  0.25
## 2  1950     43.76       43.90 -0.14
## 3  1951     52.36       52.01  0.35
## 4  1955     49.15       49.09  0.06
## 5  1959     49.33       48.43  0.90
## 6  1964     45.80       45.78  0.02
## 7  1966     46.06       46.27 -0.21
## 8  1970     45.72       46.09 -0.37
## 9  1974     42.12       42.31 -0.19
## 10 1979     47.44       47.48 -0.04
## 11 1983     47.23       47.68 -0.45
## 12 1987     46.91       46.67  0.24
## 13 1992     46.10       46.40 -0.30
## 14 1997     50.10       49.91  0.19
## 15 2001     48.91       48.09  0.82
## 16 2005     49.57       49.48  0.09
```

It seems like the difference in mean FRE scores over time with and without bootstrapping estimation is rather similar. I suppose this suggests that the bootstrapping estimate is doing a good job in predicting the FRE scores.