```
# corpus_cleaning.R

library(gutenbergr)
library(tidyverse)
library(tm)
library(tidytext)

#download books
df <- gutenberg_download(
  c(7142, 13549, 10657, 132, 13529, 46976),
  mirror = "http://mirror.csclub.uwaterloo.ca/gutenberg",
  meta_fields = c("author","title")
)

df <- df %>% mutate(
  author = case_match(
    author,
    "Sunzi, active 6th century B.C." ~ "Sun Tzu",
    "Thucydides" ~ "Thucydides",
    "Caesar, Julius" ~ "Caesar",
    "Mahan, A. T. (Alfred Thayer)" ~ "Mahan",
    "Jomini, Antoine Henri, baron de" ~ "Jomini",
    "Arrian" ~ "Arrian"
    )
)

# remove unrelated chapters and introductions by book, and group by chapter
sun_tzu <- df %>% filter(author == "Sun Tzu")
sun_tzu <- sun_tzu[
  -c(1:(which(sun_tzu$text == "Chapter I. LAYING PLANS") - 1)),
]
sun_tzu <- sun_tzu %>%
  mutate(chapter = str_detect(text, "^Chapter") %>% cumsum()) %>%
  mutate(paragraph = cumsum(text == "")) %>%
  group_by(author, title, chapter, paragraph) %>%
  summarise(text = paste(text, collapse = " ") %>% trimws()) %>%
  filter(text != "") %>%
  mutate(annotation = str_detect(text, "[\\[\\]]")) %>%
  filter(!annotation) %>% select(-annotation)

clausewitz <- read_lines("On_War.txt")
clausewitz <- tibble(
  gutenberg_id = 1946,
  text = clausewitz,
  author = "Clausewitz",
  title = "On War"
)
clausewitz <- clausewitz[-c(1:(which(clausewitz$text == "NOTICE")[2] - 1)),]
clausewitz <- clausewitz[
  -c(which(clausewitz$text == "BRIEF MEMOIR OF GENERAL CLAUSEWITZ"):
       (which(clausewitz$text == "BOOK I. ON THE NATURE OF WAR") - 1)),
  ]
clausewitz <- clausewitz %>% mutate(
  chapter = str_detect(text, "^BOOK|SKETCHES") %>% cumsum()
)

jomini <- df %>% filter(author == "Jomini")
jomini <- jomini[
  c(which(jomini$text == "DEFINITION OF THE ART OF WAR."):
      (which(jomini$text == "INDEX") - 1)),
]
jomini <- jomini %>%
  mutate(chapter = str_detect(text, "^CHAPTER") %>% cumsum())

thucydides <- df %>% filter(author == "Thucydides")
thucydides <- thucydides[-c(1:(which(thucydides$text == "BOOK I") - 1)),]
thucydides <- thucydides %>% mutate(
  chapter = str_detect(text, "^BOOK") %>% cumsum()
)

arrian <- df %>% filter(author == "Arrian")
arrian <- arrian[
  -c(1:which(arrian$text == "THE ANABASIS OF ALEXANDER.")),]
arrian <- arrian[
  c(1:(which(str_detect(arrian$text, "^FOOTNOTES")) - 1)),
]
arrian <- arrian %>% mutate(
  chapter = str_detect(text, "^BOOK") %>% cumsum()
)

mahan <- df %>% filter(author == "Mahan")
mahan <- mahan[-c(1:(which(mahan$text == "PREFACE.") - 1)),]
mahan <- mahan[-c(which(mahan$text == "CONTENTS."):
                    (which(mahan$text == "INTRODUCTORY.")[2] - 1)),]
mahan <- mahan %>% mutate(
  chapter = str_detect(text, "^CHAPTER") %>% cumsum()
)

caesar <- df %>% filter(author == "Caesar")
caesar <- caesar[
  c(which(caesar$text == "THE WAR IN GAUL")[2]:
      (which(caesar$text == "THE CIVIL WAR")[2] - 1)),]
caesar <- caesar %>% mutate(
  chapter = str_detect(text, "^BOOK") %>% cumsum()
)


# combine books and merge lines into paragraphs
books_by_line <-
  rbind(clausewitz,
        jomini,
        arrian,
        mahan,
        thucydides,
        caesar) %>%
  group_by(author) %>%
  mutate(paragraph = cumsum(text == "")) %>%
  filter(text != "") %>%
  rbind(sun_tzu)

write_csv(books_by_line, "data/book_by_line.csv")

books_by_paragraph <- books_by_line %>%
  group_by(author, title, chapter, paragraph) %>%
  summarise(text = paste(text, collapse = " ") %>% trimws()) %>%
  select(-paragraph)

write_csv(books_by_paragraph, "data/book_by_paragraph.csv")

book_by_chapter <- books_by_paragraph %>%
  group_by(author, title, chapter) %>%
  summarise(text = paste(text, collapse = " ") %>% trimws()) %>%
  filter(chapter > 0)

write_csv(book_by_chapter, "data/book_by_chapter.csv")

# clean text
df_clean <- books_by_line %>%
  mutate(
    text = text %>% stripWhitespace() %>% str_to_lower() %>%
      removePunctuation() %>% removeNumbers()
  )

#tokenize into words
tokens_words <- df_clean %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words, by = "word")

write_csv(tokens_words, "data/tokens_words_clean.csv")



# eda.R

library(tidyverse)
library(tm)
library(tidytext)
library(wordcloud)

tokens_words <- read_csv("data/tokens_words_clean.csv") %>%
  filter(!(word %in% c("footnotes", "fig", "footnote"))) %>%
  mutate(word = stemDocument(word)) %>%
  mutate(word = case_when(
    word == "citi" ~ "city",
    word == "armi" ~ "army",
    word == "alli" ~ "ally",
    word == "enemy’" ~ "enemy",
    word == "enemi" ~ "enemy",
    word == "cavalri" ~ "cavalry",
    word == "forc" ~ "force",
    word == "object" ~ "objective",
    word == "battl" ~ "battle",
    word == "posit" ~ "position",
    word == "natur" ~ "nature",
    word == "franc" ~ "france",
    word == "victori" ~ "victory",
    word == "spi" ~ "spy",
    word == "advantag" ~ "advantage",
    word == "athen" ~ "athens",
    word == "oper" ~ "operation",
    word == "alexand" ~ "alexander",
    word == "alexander’" ~ "alexander",
    word == "buonapart" ~ "buonaparte",
    word == "theatr" ~ "theatre",
    word == "theori" ~ "theory",
    word == "strategi" ~ "strategy",
    word == "organis" ~ "organisation",
    word == "maneuv" ~ "maneuver",
    word == "strateg" ~ "strategic",
    word == "artilleri" ~ "artillery",
    word == "buonapart" ~ "buonaparte",
    word == "manœuver" ~ "manœuver",
    word == "hemmedin" ~ "hemmed-in",
    word == "sunni" ~ "sunny",
    word == "ounc" ~ "ounce",
    word == "contenti" ~ "contentious",
    word == "shuaijan" ~ "shuai-jan",
    word == "argiv" ~ "argive",
    word == "hellen" ~ "hellenes",
    word == "alcibiad" ~ "alcibiades",
    TRUE ~ word
  ))

#word counts by grouped author
book_words <- tokens_words %>%
  count(author, word)

#total words per author
total_words <- book_words %>%
  group_by(author) %>%
  summarise(total_words = sum(n))

#join together
book_words <- book_words %>%
  left_join(total_words, by="author")

#token-type ratio per author
ttr_by_author <- book_words %>%
  group_by(author) %>%
  summarise(
    total_tokens = sum(n),
    unique_types = n_distinct(word), #types = words
    ttr = unique_types/total_tokens
  )

#fairly similar lexical variety, Sun Tzu is an outlier
#plot ttr by author
ttr_plot <- ttr_by_author %>%
  ggplot(aes(x = reorder(author, ttr), y = ttr, fill = author)) +
  geom_col(show.legend = FALSE) + coord_flip() +
  labs(
    title = "Type-Token Ratio by Author",
    x = "Author",
    y = "Type-Token Ratio"
  )

plot(ttr_plot)
# ggsave("images/type_token_ratio.png", ttr_plot)


#add tf-idf
books_tf_idf <- book_words %>%
  bind_tf_idf(word, author, n)

#top 10 words by term frequency per author
#initially top words for each author were numbers
top_words <- books_tf_idf %>%
  group_by(author) %>%
  slice_max(n = 10, order_by = tf) %>%
  ungroup()

#plot top 10 words by term frequency per author
top_words_plot <- top_words %>%
  ggplot(aes(x = reorder_within(word, tf, author), y = tf, fill = author)) +
           geom_col(show.legend = FALSE) +
           facet_wrap(~ author, scales = "free_y") +
           coord_flip() + scale_x_reordered() +
           labs(
             title = "Top 10 Most Frequent Words by Author",
             y = "Term Frequency",
             x = NULL)

print(top_words_plot)
# ggsave("images/tf_plot.png", top_words_plot)

#top 10 words by tf-idf per author
#initially top words for each author were numbers
top_words_tf_idf <- books_tf_idf %>%
  group_by(author) %>%
  slice_max(tf_idf, n = 10) %>%
  ungroup()

#plot top 10 words by term frequency per author
top_words_tf_idf_plot <- top_words_tf_idf %>%
  ggplot(aes(
    x = reorder_within(word, tf_idf, author),
    y = tf_idf,
    fill = author)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ author, scales = "free_y") +
  coord_flip() + scale_x_reordered() +
  labs(
    title = "Top 10 Words by TF-IDF by Author",
    y = "TF-IDF",
    x = NULL)

print(top_words_tf_idf_plot)

# can already see differences between authors, especially those focused
# on history versus those focused on tactics

create_wordcloud <- function(token_df, writer, max_words, seed) {
  set.seed(seed)
  wordcloud((token_df %>% filter(author %in% writer))$word,
            max.words = max_words)
}

create_wordcloud(tokens_words,
                 c("Sun Tzu", "Clausewitz", "Jomini"), 70, 425)
create_wordcloud(tokens_words,
                 c("Arrian", "Caesar", "Thucydides", "Mahan"), 70, 425)

create_wordcloud(tokens_words, "Sun Tzu", 50, 425)
create_wordcloud(tokens_words, "Clausewitz", 50, 425)
create_wordcloud(tokens_words, "Arrian", 50, 425)
create_wordcloud(tokens_words, "Jomini", 50, 425)
create_wordcloud(tokens_words, "Caesar", 50, 425)
create_wordcloud(tokens_words, "Mahan", 50, 425)
create_wordcloud(tokens_words, "Thucydides", 50, 425)



# lda.R

library(tidyverse)
library(tm)
library(tidytext)
library(topicmodels)

books <- read_csv("data/book_by_chapter.csv")

book_dtm <- books %>% group_by(author, chapter) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words, by = join_by(word)) %>%
  group_by(author, chapter, word)%>% count() %>%
  mutate(book_chapter = paste(author, chapter, sep="_")) %>%
  cast_dtm(book_chapter, word, n)

trials <- 10
mismatch <-
  tibble(chapter = character(),
         topic = numeric(),
         author = character())
duplicate <- c()
for (seed in c(1:trials)) {
  book_lda <- book_dtm %>% LDA(k = 7, control = list(seed = seed))

  book_lda_tidy <- book_lda %>% tidy() %>% arrange(desc(beta))
  book_lda_tidy_gamma <- book_lda %>% tidy(matrix = "gamma") %>%
    separate(document, c("author", "chapter"), sep = "_")

  consensus <- book_lda_tidy_gamma %>% group_by(author, chapter) %>%
    arrange(desc(gamma)) %>% slice_max(gamma, n = 1) %>% group_by(author) %>%
    count(topic) %>% slice_max(n, n = 1) %>% select(-n)

  chapter_mismatch <- book_lda_tidy_gamma %>% group_by(author, chapter) %>%
    arrange(desc(gamma)) %>% slice_max(gamma, n = 1) %>%
    anti_join(consensus, by = join_by(author, topic))

  chapter_topic <- book_lda_tidy_gamma %>% group_by(author, chapter) %>%
    slice_max(gamma, n = 1)

  pairs <- consensus %>% ungroup() %>% group_by(topic) %>% filter(n() > 1) %>%
    summarise(authors = paste(author, collapse = " & "))

  for (i in pairs$authors) {
    if (!(i %in% names(duplicate))) {
      duplicate[i] <- 1
    }
    else {
      duplicate[i] <- duplicate[i] + 1
    }
  }

  mismatch <-
    mismatch %>% bind_rows(
      chapter_mismatch %>% mutate(chapter = paste(author, chapter)) %>%
        ungroup() %>% select(chapter, topic) %>%
        left_join(consensus, by = join_by(topic))
    )
}

same_topic_plot <- tibble(authors = names(duplicate), n = duplicate) %>%
  ggplot(aes(fct_reorder(authors, n), n)) + geom_col() + coord_flip() +
  labs(
    title = "Number of Times Authors Classified Into Same Topic",
    y = "",
    x = "Pairs of Authors")
plot(same_topic_plot)

misclassification_plot <-
  mismatch %>% replace_na(list(author = "none")) %>%
  group_by(chapter, author) %>% count() %>%
  ggplot(aes(reorder_within(chapter, n, author), n, fill = author)) +
  geom_col(show.legend = FALSE) + coord_flip() +
  scale_x_reordered() + facet_wrap(~ author, scales = "free_y") +
  labs(title = "Misclassified Chapters by Author Topic",
       x = "Chapter", y = "Count")
plot(misclassification_plot)



# sentiment.R

library(tidyverse)
library(tm)
library(tidytext)

book <- read_csv("data/book_by_chapter.csv")

tokens_words <- book %>%
  mutate(
    text = text %>% stripWhitespace() %>% str_to_lower() %>%
      removePunctuation() %>% removeNumbers(),
    chapter = as.factor(chapter)
  ) %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words, by = "word")

bing <- tokens_words %>%
  inner_join(get_sentiments("bing"), by = join_by(word)) %>%
  group_by(author, chapter)

bing_mean <- bing %>% count(sentiment) %>%
  mutate(prop = n / sum(n)) %>% filter(sentiment == "negative") %>%
  group_by(author) %>%
  summarise(mean = mean(prop), sd = sd(prop))

bing %>% group_by(author) %>% count(sentiment) %>%
  mutate(prop = n / sum(n)) %>% filter(sentiment == "negative") %>%
  ggplot(aes(author, prop)) +
  geom_col(show.legend = FALSE) +
  geom_hline(aes(yintercept = mean(prop))) +
  labs(title = "Proportion of Negative Sentiment by Author (Bing)",
       x = "Author", y = "Negative Proportion")

bing %>% count(sentiment) %>%
  mutate(prop = n / sum(n)) %>% filter(sentiment == "negative") %>%
  ggplot(aes(chapter, prop)) +
  geom_col(show.legend = FALSE) + facet_wrap(~ author, scales = "free_y") +
  scale_x_reordered() + coord_flip() +
  geom_hline(aes(yintercept = mean, col = "Mean"), data = bing_mean) +
  geom_hline(aes(yintercept = mean - 2 * sd, col = "-2SD"),
             data = bing_mean) +
  labs(title = "Proportion of Negative Sentiment by Chapter (Bing)",
       x = "Chapter", y = "Negative Proportion",
       col = "")

afinn <- tokens_words %>%
  inner_join(get_sentiments("afinn"), by = join_by(word))

afinn %>%
  ggplot(aes(author, value)) +
  geom_boxplot(show.legend = FALSE) +
  labs(title = "Sentiment Level by Author (AFINN)",
       x = "Author", y = "Sentiment Level",
       col = "")

afinn %>%
  ggplot(aes(chapter, value)) +
  geom_boxplot(show.legend = FALSE) +
  facet_wrap(~ author, scales = "free_y") +
  scale_x_reordered() +
  coord_flip() + geom_hline(yintercept = 0) +
  labs(title = "Sentiment Level by Chapter (AFINN)",
       x = "Chapter", y = "Sentiment Level",
       col = "")

nrc <- tokens_words %>%
  inner_join(get_sentiments("nrc"), by = join_by(word))

nrc %>% group_by(author) %>% count(sentiment) %>%
  filter(sentiment %in% c("positive", "negative")) %>%
  mutate(prop = n / sum(n)) %>%
  filter(sentiment == "negative") %>%
  ggplot(aes(author, prop)) +
  geom_col(show.legend = FALSE) +
  geom_hline(aes(yintercept = 0.5)) +
  labs(title = "Proportion of Negative Sentiment by Author (NRC)",
       x = "Author", y = "Negative Proportion")

nrc %>% group_by(author) %>% count(sentiment) %>%
  filter(!(sentiment %in% c("positive", "negative"))) %>%
  mutate(prop = n / sum(n)) %>%
  ggplot(aes(reorder_within(sentiment, n, author), prop, fill = sentiment)) +
  coord_flip() +scale_x_reordered() +
  geom_col(show.legend = FALSE) + facet_wrap(~ author, scales = "free_y") +
  geom_hline(aes(yintercept = 1/8)) +
  labs(title = "Proportion of Emotions by Author (NRC)",
       x = "Emotion", y = "Proportion")



# ngrams.R

library(tidyverse)
library(tm)
library(tidytext)
library(igraph)
library(ggraph)

books <- read_csv("data/book_by_chapter.csv")

jomini <- books %>% filter(author == "Jomini")
jomini$text[7] <- strsplit(jomini$text[7], "")[[1]][
  -c(unlist(gregexpr("_Different Formations", jomini$text[7])):
       (unlist(gregexpr("Note.--In all these", jomini$text[7])) - 1))
] %>% paste(collapse = "")
jomini <- jomini %>%
  mutate(text = str_remove_all(text, "FOOTNOTES:|\\[Illustration[^\\]]*\\]"))

books <- rbind(book %>% filter(author == "Clausewitz"), jomini)

book_bigrams <- books %>% unnest_tokens(bigram, text, token = "ngrams", n = 2)
```

```
bigram_count <- book_bigrams %>% group_by(author) %>% count(bigram)
bigram_count <- bigram_count %>%
  separate(bigram, c("word1", "word2"), sep = " ")
bigram_count <- bigram_count %>%
  anti_join(stop_words, by = join_by(word1 == word)) %>%
  anti_join(stop_words, by = join_by(word2 == word)) %>%
  mutate(bigram = paste(word1, word2))

bigram_tf_idf <- bigram_count %>% bind_tf_idf(bigram, author, n)

bigram_tf_idf %>% group_by(author) %>% slice_max(tf, n = 10)  %>%
  ggplot(aes(reorder_within(bigram, tf, author), tf, fill = author)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ author, scales = "free_y") +
  coord_flip() + scale_x_reordered()

bigram_tf_idf %>% group_by(author) %>% slice_max(tf_idf, n = 10)  %>%
  ggplot(aes(reorder_within(bigram, tf_idf, author), tf_idf, fill = author)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ author, scales = "free_y") +
  coord_flip() + scale_x_reordered()

most_frequent_bigrams <- bigram_count %>% group_by(author) %>%
  slice_max(n, n = 30) %>% ungroup()

set.seed(425)
most_frequent_bigrams %>% filter(author == "Clausewitz") %>%
  select(word1, word2) %>% graph_from_data_frame() %>%
  ggraph(layout = "fr") + geom_edge_link() + geom_node_point() +
  geom_node_text(aes(label = name), vjust = 1)

set.seed(425)
most_frequent_bigrams %>% filter(author == "Jomini") %>%
  select(word1, word2) %>% graph_from_data_frame() %>%
  ggraph(layout = "fr") + geom_edge_link() + geom_node_point() +
  geom_node_text(aes(label = name), vjust = 1)
```