Lab 08 - Text Mining/NLP

Learning goals

- Use `unnest_tokens()` and `unnest_ngrams()` to extract tokens and ngrams from text
- Use dplyr and ggplot2 to analyze and visualize text data
- Try a theme model using `topicmodels`

Lab description

For this lab we will be working with the medical record transcriptions from https://www.mtsamples.com/ available at https://github.com/JSC370/JSC370-2025/tree/main/data/medical_transcriptions.

Deliverables

1. Questions 1-7 answered, knit to pdf or html output uploaded to Quercus.
2. Render the Rmarkdown document using `github_document` and add it to your github site. Add link to github site in your html.

Setup packages

You should load in `tidyverse`, (or `data.table`), `tidytext`, `wordcloud2`, `tm`, and `topicmodels`.

Read in the Medical Transcriptions

Loading in reference transcription samples from https://www.mtsamples.com/

```
library(tidytext)
library(tidyverse)
library(wordcloud2)
library(tm)
library(topicmodels)

mt_samples <- read_csv("https://raw.githubusercontent.com/JSC370/JSC370-2025/main/data/medical_transcrip
mt_samples <- mt_samples |>
  select(description, medical_specialty, transcription)

head(mt_samples)
```

Question 1: What specialties do we have?

We can use `count()` from `dplyr` to figure out how many different medical specialties are in the data. Are these categories related? overlapping? evenly distributed? Make a bar plot.

```
mt_samples |>
  count(medical_specialty, sort = TRUE) |>
  ggplot(aes(fct_reorder(medical_specialty,n),n)) +
  geom_col(fill = "dodgerblue") +
  coord_flip() +
  theme_bw()
```

— The medical specialties are related but vary in focus, with some overlap (e.g., Neurology & Neurosurgery). The distribution is highly skewed, with Surgery being the most common and some specialties having very few occurrences.

Question 2: Tokenize

- Tokenize the the words in the `transcription` column
- Count the number of times each token appears
- Visualize the top 20 most frequent words with a bar plot
- Create a word cloud of the top 20 most frequent words

Explain what we see from this result. Does it makes sense? What insights (if any) do we get?

```
tokens <- mt_samples |>
  select(transcription) |>
  unnest_tokens(word, transcription) |>
  group_by(word) |>
  summarize(word_frequency = n()) |>
  arrange(across(word_frequency, desc)) |>
  head(20)

tokens |>
  ggplot(aes(fct_reorder(word, word_frequency), word_frequency)) +
  geom_bar(stat = "identity", fill = "dodgerblue") +
  coord_flip() +
  theme_bw()

tokens |>
  count(word, sort = TRUE) |>
  wordcloud2(size = 0.5, color = "random-light", backgroundColor = "dodgerblue")
```

— The results show that common stopwords dominate, with “patient” being the only notable medical term. The bar plot provides clear frequency insights, while the word cloud highlights key words visually. Removing stopwords could reveal more meaningful medical terms.

Question 3: Stopwords

- Redo Question 2 but remove stopwords

- Check `stopwords()` library and `stop_words` in `tidytext`
- Use regex to remove numbers as well
- Try customizing your stopwords list to include 3-4 additional words that do not appear informative

What do we see when you remove stopwords and then when you filter further? Does it give us a better idea of what the text is about?

```
head(stopwords("english"))
length(stopwords("english"))
head(stop_words)

tokens2 <- mt_samples |>
  select(transcription) |>
  unnest_tokens(word, transcription, token = "words") |>
  anti_join(stop_words, by = "word") |>
  filter(!str_detect(word, "^[0-9]+$")) |> #[[:digit:]]+
  filter(!word %in% c("mm", "mg", "noted")) |>
  count(word, sort = TRUE) |>
  top_n(20, n)

tokens2 |>
  ggplot(aes(n, fct_reorder(word, n))) +
  geom_col(fill = "dodgerblue") +
  theme_bw()

tokens2 |>
  count(word, sort = TRUE) |>
  wordcloud2(size = 0.4, color = "random-light", backgroundColor = "dodgerblue")
```

— After removing stopwords, the analysis reveals more medical terms, providing clearer insight into the content. Further filtering enhances relevance, helping identify key themes related to medical procedures and patient care.

Question 4: ngrams

Repeat question 2, but this time tokenize into bi-grams. How does the result change if you look at tri-grams? Note we need to remove stopwords a little differently. You don't need to recreate the wordclouds.

```
stop_words2 <- c("en", "mm", "mg", "count", stop_words$word)
sw_start <- paste0("^", paste(stop_words2, collapse=" |^"), "$")
sw_end <- paste0("", paste(stop_words2, collapse="$| "), "$")

tokens_bigram <- mt_samples |>
  select(transcription) |>
  unnest_tokens(ngram, transcription, token = "ngrams", n = 2) |>
  filter(!grepl(sw_start, ngram, ignore.case = TRUE))|>
  filter(!grepl(sw_end, ngram, ignore.case = TRUE))|>
  filter(!grepl("[[:digit:]]+", ngram))|>
  group_by(ngram) %>%
  summarize(word_frequency = n()) %>%
  arrange(across(word_frequency, desc)) %>%
```

```
  head(20)

tokens_bigram %>%
  ggplot(aes(ngram, word_frequency)) +
  geom_col(fill="dodgerblue") +
  coord_flip() +
  theme_bw()
```

— The most frequent bi-grams, such as "preoperative diagnosis," "postoperative diagnosis," and "blood pressure," highlight key medical procedures, patient conditions, and documentation phrases commonly used in clinical settings.

Question 5: Examining words

Using the results from the bigram, pick a word and count the words that appear before and after it, and create a plot of the top 20.

```
library(stringr)
# e.g. patient, blood, preoperative...
tokens_bigram |>
  filter(str_detect(ngram, regex("\\sblood$|^blood\\s"))) |>
  mutate(word = str_remove(ngram, "blood"),
         word = str_remove_all(word, " ")) |>
  group_by(word) |>
  head(20) |>
  ggplot(aes(reorder(word, word_frequency), word_frequency)) +
  geom_col(fill = "dodgerblue") +
  theme_bw()
```

— The analysis of bi-grams containing "blood" highlights common medical terms like "estimated," "pressure," and "loss," reflecting key clinical concerns related to blood measurements, conditions, and assessments in medical documentation.

Question 6: Words by Specialties

Which words are most used in each of the specialties? You can use `group_by()` and `top_n()` from `dplyr` to have the calculations be done within each specialty. Remember to remove stopwords. How about the 5 most used words?

```
mt_samples |>
   unnest_tokens(word, transcription) |>
   anti_join(stop_words, by = "word") |>
   filter(!str_detect(word, "^[0-9]+$")) |>
   filter(!word %in% c("mm","mg","noted")) |>
   group_by(medical_specialty) |>
   count(word,sort = TRUE)|>
   top_n(1,n)

mt_samples |>
   unnest_tokens(word, transcription) |>
   anti_join(stop_words, by = "word") |>
```

```
  filter(!str_detect(word, "^[0-9]+$")) |>
  filter(!word %in% c("mm","mg","noted")) |>
  group_by(medical_specialty) |>
  count(word,sort = TRUE)|>
  top_n(5,n)
```

— The results make sense, with "patient" dominating most specialties, while specialties like Radiology and Neurology frequently use "left".

Question 7: Topic Models

See if there are any themes in the data by using a topic model (LDA).

- you first need to create a document term matrix
- then you can try the LDA function in `topicmodels`. Try different k values.
- create a facet plot of the results from the LDA (see code from lecture)

```
transcripts_dtm <- mt_samples |>
  select(transcription) |>
  unnest_tokens(word, transcription) |>
  anti_join(stop_words, by = "word") |>
  filter(!str_detect(word, "^[0-9]+$")) |> #[[:digit:]]+
  filter(!word %in% c("mm", "mg", "noted")) |>
  DocumentTermMatrix()

transcripts_dtm <- as.matrix(transcripts_dtm)

transcripts_lda1 <- LDA(transcripts_dtm, k = 5,
                        control = list(seed = 1234))

transcripts_lda2 <- LDA(transcripts_dtm, k = 3,
                        control = list(seed = 1234))

transcripts_top_terms1 <-
  tidy(transcripts_lda1, matrix = "beta") |>
  filter(!str_detect(term, "^[0-9]+$")) |>
  group_by(topic) |>
  slice_max(beta, n = 10) |>
  ungroup() |>
  arrange(topic, -beta)

transcripts_top_terms2 <-
  tidy(transcripts_lda2, matrix = "beta") |>
  filter(!str_detect(term, "^[0-9]+$")) |>
  group_by(topic) |>
  slice_max(beta, n = 10) |>
  ungroup() |>
  arrange(topic, -beta)

transcripts_top_terms1 |>
  mutate(term = reorder_within(term, beta, topic)) |>
  ggplot(aes(beta, term, fill = factor(topic))) +
```

```
  geom_col(show.legend = FALSE) +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered() +
  theme_bw()

transcripts_top_terms2 |>
  mutate(term = reorder_within(term, beta, topic)) |>
  ggplot(aes(beta, term, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~topic, scales = "free") +
  scale_y_reordered() +
  theme_bw()
```