

Pre-process the speeches in R

BA3 Text as Data · Session 2 · BA2 review track (R)

Due Oct. 2 at 09:15 in your class/project repository: demos/02-preprocessing/. I am asking you to create one R script, three word clouds, and a short reflection (.md or .txt README file is fine). Follow the steps in order and add each code block to your script.

Before you start

Download these two files from the [Session 2](#) page into your data/ folder.

- `presidential-speeches-100.csv`, the raw speeches
- `stopwords-ko.txt`, the course stopwords list

Step 1 · Create the script

In your course RStudio Project, choose File > New File > R Script. Save it as `demos/02-preprocessing/workflow/s02-preprocessing.R`

```
# BA3 Text as Data · Session 2 · Preprocess the presidential speeches
```

Step 2 · Install and load the packages

```
packages <- c("readr", "dplyr", "tidyr", "reticulate", "ggplot2", "ggwordcloud",  
  ↪ "ragg")  
install.packages(setdiff(packages, rownames(installed.packages())))  
  
library(readr)  
library(dplyr)  
library(tidyr)  
library(reticulate)  
library(ggplot2)  
library(ggwordcloud)
```

Step 3 · Set a seed

If you set a seed, that means any word cloud you produce will be produced in precisely the way that you have done it in your script. This is for reproducibility purposes. It is good practice and FAIR.

```
set.seed(2026)
```

Step 4 · Import the data

Similar to Step 3, a relative path and named column types let anyone repeat the same file import you did.

```
speeches <- read_csv(  
  "data/presidential-speeches-100.csv",  
  col_types = cols(.default = col_character(), division_number = col_integer())  
)  
stopwords_ko <- read_lines("data/stopwords-ko.txt")  
  
nrow(speeches)      # 100  
count(speeches, kind) # speech types
```

In the data/README.md (or .txt) record the file name, the [corpus card](#) (from the course website) as its source, and the download date. I do not consider this totally necessary, but I do think it is good practice, and I encourage you to do it anyway.

Step 5 • Split the speeches into morphemes with Kiwi

Kiwi is the Korean analyzer from class. The reticulate package runs it in and from R. The first time, `py_require()` downloads Kiwi, which takes a few minutes. You might have to debug here.

```
py_require("kiwipiepy")
kiwi <- import("kiwipiepy")$Kiwi()

kiwi_tokens <- function(text) {
  tokens <- kiwi$tokenize(text)
  tibble(word = vapply(tokens, function(t) t$form, ""),
         pos = vapply(tokens, function(t) t$tag, ""))
}

morphemes <- speeches |>
  mutate(tokens = lapply(speech_text, kiwi_tokens)) |>
  select(division_number, president, kind, tokens) |>
  unnest(tokens)

head(morphemes, 20)      # one row per morpheme, with its POS tag
```

Step 6 • Pre-process

Each line below is one pre-processing operation that we reviewed in class (and you have done before in ODM).

```
keep_pos <- c("NNG", "NNP")      # POS tags to keep: nouns
min_length <- 2                  # shortest token to keep

clean <- morphemes |>
  mutate(pos = sub("-.*", "", pos)) |>      # VV-R becomes VV
  filter(pos %in% keep_pos) |>             # part-of-speech filter
  filter(nchar(word) >= min_length) |>     # length filter
  filter(!grepl("[0-9]+$", word)) |>       # drop numbers
  filter(!word %in% stopwords_ko)         # drop stopwords

nrow(clean)                        # about 28,850
```

To tweak anything here, change a setting and run Step 6 again. Add "VV", "VA" to `keep_pos` to keep verbs and adjectives, or put # before the stopwords line to switch it off. Make sure you basically understand the lines above. You will reuse them.

Step 7 • Word cloud of all speeches

```
korean_font <- switch(Sys.info()[["sysname"]],
  Windows = "Malgun Gothic", Darwin = "AppleGothic", "Noto Sans CJK KR")

dir.create("demos/02-preprocessing/output", showWarnings = FALSE)

cloud_all <- clean |>
  count(word, sort = TRUE) |>
  slice_head(n = 80) |>
  ggplot(aes(label = word, size = n)) +
  geom_text_wordcloud(family = korean_font, eccentricity = 1) +
  scale_size_area(max_size = 12) +
  labs(title = "All speeches") +
  theme_minimal(base_family = korean_font)

ggsave("demos/02-preprocessing/output/wordcloud-all.png", cloud_all,
```

```
device = ragg::agg_png, width = 8, height = 5, dpi = 200, bg = "white")
```

Step 8 • Commemorative speeches (기념사)

```
cloud_commemorative <- clean |>
  filter(kind == "기념사") |>
  count(word, sort = TRUE) |>
  slice_head(n = 80) |>
  ggplot(aes(label = word, size = n)) +
  geom_text_wordcloud(family = korean_font, eccentricity = 1) +
  scale_size_area(max_size = 12) +
  labs(title = "Commemorative speeches (기념사)") +
  theme_minimal(base_family = korean_font)

ggsave("demos/02-preprocessing/output/wordcloud-commemorative.png",
  cloud_commemorative,
  device = ragg::agg_png, width = 8, height = 5, dpi = 200, bg = "white")
```

Step 9 • National Assembly speeches (국회연설)

Copy Step 8 and change the object name to `cloud_assembly`, the filter to `kind == "국회연설"`, the title, and the file name to `wordcloud-national-assembly.png`.

Step 10 • Record your R setup

```
writelines(capture.output(sessionInfo()),
  "demos/02-preprocessing/output/session-info.txt")
```

Save the script and run it again from top to bottom (Ctrl+Shift+Enter or Cmd+Shift+Enter) to see if it works.

Step 11 • Reflect

In `demos/02-preprocessing/README.md`, list the settings you used in Step 6 and write a few sentences on what you did and what you found in the three word clouds. Nothing fancy, just describe it as you see it.

Submit

The `.R` script, the three PNG files in `output/`, and the `README`. Commit and push to the GitHub repository.

If Korean shows as boxes in the Plots pane, the saved PNG files are still correct. To fix the pane, choose Tools, Global Options, General, Graphics, Backend AGG. You might have to debug this a little to get it to work.