Word Embeddings of Congress

The output is a data frame with the top words for each keyword (“climate”, “woman”, “environmentalist”, and “government”)

```
# ---- Libraries ----
library(text2vec)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readr)
library(stringr)
library(lubridate)
library(tibble)
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
##
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
# ---- Config ----
path <- "~/Desktop/TMHA/congress_data_daily_by_speaker_with_metadata_2000-2020_climate-change.csv"
keyword <- "climate"
top_n   <- 10

# ---- Load + clean + get year/decade ----
df <- read_csv(path, show_col_types = FALSE) %>%
  mutate(content = as.character(content)) %>%
  filter(!is.na(content), str_squish(content) != "")

# Get a year column if not present
if (!"year" %in% names(df)) {
```

```r
  if ("date" %in% names(df)) {
    df <- df %>% mutate(year = year(suppressWarnings(ymd(date))))
  } else if ("speech_date" %in% names(df)) {
    df <- df %>% mutate(year = year(suppressWarnings(ymd(speech_date))))
  } else {
    stop("No 'year' or parseable date column found. Please add a 'year' column.")
  }
}

df <- df %>%
  mutate(
    year   = as.integer(year),
    decade = case_when(
      year >= 2000 & year < 2010 ~ "2000",
      year >= 2010 & year < 2020 ~ "2010",
      year >= 2020 & year < 2030 ~ "2020",
      TRUE ~ NA_character_
    )
  ) %>%
  filter(!is.na(decade))

# ---- Helper: most-similar words from a vector of texts ----
most_similar_from_text <- function(text_vec, keyword = "climate", top_n = 10,
                                   rank = 50, window = 5L, term_count_min = 1,
                                   x_max = 10, n_iter = 30, seed = 42) {

  text_vec <- text_vec[!is.na(text_vec) & str_squish(text_vec) != ""]
  if (length(text_vec) < 1) {
    return(tibble(word = "No text available in this subset.", similarity = NA_real_))
  }

  tokens <- word_tokenizer(str_to_lower(text_vec))
  it <- itoken(tokens, ids = seq_along(tokens), progressbar = FALSE)

  vocab <- create_vocabulary(it) %>%
    prune_vocabulary(term_count_min = term_count_min)

  if (nrow(vocab) == 0) {
    return(tibble(word = "Vocabulary is empty for this subset.", similarity = NA_real_))
  }

  vectorizer <- vocab_vectorizer(vocab)
  tcm <- create_tcm(it, vectorizer, skip_grams_window = window)

  if (length(tcm@x) == 0) {
    return(tibble(word = "TCM is empty (no co-occurrences).", similarity = NA_real_))
    }

  set.seed(seed)
  glove <- GlobalVectors$new(rank = rank, x_max = x_max)
  wv_main <- glove$fit_transform(tcm, n_iter = n_iter)
  wv_context <- glove$components
  word_embeddings <- wv_main + t(wv_context)
```

```
  if (!keyword %in% rownames(word_embeddings)) {
    return(tibble(
      word = sprintf("Keyword '%s' not in vocabulary (lower term_count_min or change keyword).", keyword),
      similarity = NA_real_
    ))
  }

  kw_vec <- word_embeddings[keyword, , drop = FALSE]
  cos_sim <- sim2(word_embeddings, kw_vec, method = "cosine", norm = "l2")[, 1]

  tibble(
    word = names(cos_sim),
    similarity = unname(cos_sim)
  ) %>%
    filter(word != keyword) %>%
    arrange(desc(similarity)) %>%
    slice_head(n = top_n)
}

# ---- Compute per-decade tables ----
most_similar_2000 <- df %>%
  filter(decade == "2000") %>%
  { most_similar_from_text(.$content, keyword = keyword, top_n = top_n) }

most_similar_2010 <- df %>%
  filter(decade == "2010") %>%
  { most_similar_from_text(.$content, keyword = keyword, top_n = top_n) }

most_similar_2020 <- df %>%
  filter(decade == "2020") %>%
  { most_similar_from_text(.$content, keyword = keyword, top_n = top_n) }

# ---- Render tables (kable) ----
tbl_2000 <- most_similar_2000 %>%
  mutate(similarity = ifelse(is.na(similarity), NA, round(similarity, 3))) %>%
  kable(
    caption   = 'Words Most Related to "Climate": 2000s Congressional Records',
    col.names = c("Word", "Cosine similarity"),
    align     = c("l", "r"),
    booktabs  = TRUE
  ) %>%
  kable_styling(full_width = FALSE)
```

```
tbl_2010 <- most_similar_2010 %>%
  mutate(similarity = ifelse(is.na(similarity), NA, round(similarity, 3))) %>%
  kable(
    caption   = 'Words Most Related to "Climate": 2010s Congressional Records',
    col.names = c("Word", "Cosine similarity"),
    align     = c("l", "r"),
    booktabs  = TRUE
  ) %>%
  kable_styling(full_width = FALSE)
```

```r
tbl_2020 <- most_similar_2020 %>%
  mutate(similarity = ifelse(is.na(similarity), NA, round(similarity, 3))) %>%
  kable(
    caption   = 'Words Most Related to "Climate": 2020s Congressional Records',
    col.names = c("Word", "Cosine similarity"),
    align     = c("l", "r"),
    booktabs  = TRUE) %>%
  kable_styling(full_width = FALSE)
```

```r
bl1 <- most_similar_df_1 %>%
  mutate(similarity = round(similarity, 3)) %>%
  kable(
    caption   = 'Words Most Related to "Climate": Searching the 2001 Congressional Records',
    col.names = c("Word", "Cosine similarity"),
    align     = c("l", "r"),
    booktabs  = TRUE
  ) %>%
  kable_styling(full_width = FALSE)

tbl2 <- most_similar_df_2 %>%
  mutate(similarity = round(similarity, 3)) %>%
  kable(
    caption   = 'Words Most Related to "Climate": Searching the 2021 Congressional Records',
    col.names = c("Word", "Cosine similarity"),
    align     = c("l", "r"),
    booktabs  = TRUE
  ) %>%
  kable_styling(full_width = FALSE)

kables(list(tbl1, tbl2)) %>%
  kable_styling()
```