# ==== Setup ====

install.packages(c("tidyverse", "tidytext", "readtext", "textclean", "textdata",

"widyr", "igraph", "ggraph", "tm", "topicmodels"))

library(tidyverse)

library(tidytext)

library(readtext)

library(textclean)

library(textdata)

library(widyr)

library(igraph)

library(ggraph)

library(tm)

library(topicmodels)

# ==== 1. Read and preprocess ====

file\_path <- file.choose()

raw\_text <- readtext(file\_path, encoding = "UTF-8")$text

clean\_text <- raw\_text %>%

replace\_non\_ascii() %>%

replace\_html() %>%

replace\_contraction() %>%

replace\_emoji() %>%

tolower() %>%

str\_replace\_all("[0-9]+", "") %>%

str\_replace\_all("[^a-z\\s]", "") %>%

str\_squish()

# ==== 2. Convert to tibble and tokenize ====

text\_df <- tibble(line = 1, text = clean\_text) %>%

unnest\_tokens(word, text)

# ==== 3. Remove ultra-common English stopwords and extras ====

data("stop\_words")

extra\_stopwords <- c(

# domain-specific junk

"iran", "afghanistan", "afghan", "unhcr", "grandi",

"kazemi", "minister", "refugees", "foreign", "national", "nationals",

"islamic", "republic", "government", "tehran", "education",

"students", "school", "service", "services", "officials", "department",

"meeting", "meeting", "support", "program", "platform", "certificate",

"countries", "country", "people", "children", "year", "years",

"iranian", "reported", "statement", "according", "report", "website",

"khabar", "online", "wrote", "news", "agency", "based"

)

all\_stops <- bind\_rows(stop\_words, tibble(word = extra\_stopwords, lexicon = "custom"))

text\_df <- text\_df %>%

anti\_join(all\_stops, by = "word") %>%

filter(str\_detect(word, "^[a-z]+$")) # only words, no numbers/punctuation

# ==== 4. Word frequency ====

text\_df %>%

count(word, sort = TRUE) %>%

top\_n(25) %>%

ggplot(aes(x = reorder(word, n), y = n)) +

geom\_col(fill = "darkgreen") +

coord\_flip() +

labs(title = "Top 25 Most Frequent Words (After Deep Cleaning)")

# 5. Sentiment analysis (Bing lexicon)

bing\_sentiment <- text\_df %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort = TRUE)

# Plot sentiment

bing\_sentiment %>%

group\_by(sentiment) %>%

top\_n(10, n) %>%

ggplot(aes(reorder(word, n), n, fill = sentiment)) +

geom\_col(show.legend = FALSE) +

facet\_wrap(~ sentiment, scales = "free\_y") +

coord\_flip() +

labs(title = "Sentiment Analysis", x = "Word", y = "Frequency")

# فرض: text\_df شامل ستون word هست و از قبل پاک‌سازی شده

# بخش‌بندی متن به قسمت‌های مصنوعی (مثلاً هر 50 خط یا جمله)

text\_df <- text\_df %>%

mutate(section = row\_number() %/% 50)

# محاسبه هم‌واژگانی

library(widyr)

pair\_counts <- text\_df %>%

pairwise\_count(word, section, sort = TRUE)

# نمایش 20 زوج‌واژه‌ی پرتکرار

top\_20\_pairs <- pair\_counts %>%

slice\_max(n, n = 20)

# نمایش جدول

print(top\_20\_pairs)

# ==== 7. Topic Modeling (LDA) ====

dtm <- text\_df %>%

count(section, word) %>%

cast\_dtm(section, word, n)

lda\_model <- LDA(dtm, k = 2, control = list(seed = 1234))

topics <- tidy(lda\_model, matrix = "beta")

top\_terms <- topics %>%

group\_by(topic) %>%

top\_n(8, beta) %>%

ungroup() %>%

arrange(topic, -beta)

top\_terms %>%

ggplot(aes(reorder\_within(term, beta, topic), beta, fill = factor(topic))) +

geom\_col(show.legend = FALSE) +

facet\_wrap(~ topic, scales = "free\_y") +

scale\_x\_reordered() +

coord\_flip() +

labs(title = "Top Words per Topic", x = "Term", y = "Beta")