AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Spring 2024-2025

Course Title: **Introduction to Data Science**

Course Teacher: **DR. ABDUS SALAM**

Section: **E**

Group No. **3**

Serial Number	Name	ID
01.	Md. Ashfaq Bin Hoque	22-46662-1
02.	Md. Deniad Alam	22-46658-1
03.	Maruf Billah Siddiki	22-47177-1
04.	Al Fahim	22-46402-1

Text Preprocessing Steps

In this project, raw news article content was preprocessed using a series of natural language processing (NLP) steps to prepare the data for analysis, such as topic modeling. The steps included:

- Emoji Replacement: Converted emojis to text using replace_emoji().
- Contraction Expansion: Replaced contractions (e.g., "don't" → "do not") using replace_contraction().
- Lowercasing: Converted all text to lowercase.
- Dash Replacement: Replaced hyphens with spaces.
- Punctuation Removal: Removed all punctuation marks.
- Whitespace Normalization: Removed extra spaces.
- Trimming: Removed leading and trailing spaces.
- Tokenization: Split text into individual words.
- Stopword Removal: Removed common English stopwords.
- Lemmatization: Reduced words to their root form.
- Numeric Token Removal: Removed words containing digits.
- Reconstruction: Joined tokens back into clean text.

Topic Modeling Steps

In this workflow, Latent Dirichlet Allocation (LDA) was applied to identify latent topics in the cleaned news dataset. The key steps are as follows:

- Data Preparation:
  Loaded cleaned news content from CSV for analysis.
- Creating Document-Term Matrix (DTM):
  Converted text corpus into a matrix of word counts per document.
- Topic Modeling with LDA:
  Applied Latent Dirichlet Allocation to discover hidden topics in the text.
- Extracting top terms per topic:
  Selected the 10 most representative words for each topic using term probabilities.
- Get the most probable words for each topic:
  Identified top keywords that best describe each topic based on highest beta values.
- Get the topic proportions for each document:
  Calculated how much each topic contributes to individual documents using gamma values.
- Interpret the Results:
  Analyzed top words and document-topic distributions to understand the themes in the news articles.

PART 1: WEB SCRAPING AND PREPROCESSING

PART 1.1: WEB SCRAPING

1.1.1 Load Required Libraries

Description:

This section loads the necessary R packages used throughout the script. The rvest library is used for web scraping by extracting content from HTML nodes. dplyr is used for data manipulation and transformation. RSelenium enables browser automation, allowing interaction with JavaScript-driven websites such as clicking buttons or scrolling. wdman is used to manage WebDriver binaries like GeckoDriver for Firefox. Lastly, netstat helps identify and allocate a free port on the local machine to run the Selenium server.

Code:

```
library(rvest)
library(dplyr)
library(RSelenium)
library(wdman)
library(netstat)

```

Output:

All libraries are loaded with no errors.

1.1.2 Define clickAjaxButton() Function

Description:

The clickAjaxButton() function simulates clicking an AJAX-powered "Load More" button multiple times to reveal additional articles on a webpage. It uses RSelenium's findElement to locate the button via its CSS selector, and clickElement to perform the click. A delay is introduced using Sys.sleep() after each click to allow new content to load before the next interaction. This function is essential when scraping websites that load data dynamically instead of serving it all at once.

Code:

```r
clickAjaxButton <- function(times, button_selector, delay = 5000) {
  for (i in 1:times) {
    button <- remote_driver$findElement(using = "css selector", paste0("", button_selector))

    if (!is.null(button)) {
      button$clickElement()
      Sys.sleep(delay / 1000)
    } else {
      cat("Button not found!\n")
      break
    }
  }
}
```

Output:

AJAX-loaded articles are revealed in the browser session.

1.1.3 Define Content Scraping Helper Functions

Description:

Three helper functions—get_title(), get_content(), and get_date()—are defined to extract specific elements from an individual article's web page. Each function uses read_html() to load the HTML content of the page and then utilizes html_nodes() and html_text() to extract the relevant text. For example, get_title() pulls the headline, get_content() concatenates all paragraph tags into a full article body, and get_date() extracts and cleans the publish date by removing prefixes like "Published:". These functions modularize the scraping process and make the main function more readable.

Code:

```r
get_title = function(inside_link){
  inside_page = read_html(inside_link)
  title = inside_page %>%  html_nodes(".mb10") %>% html_text()
  return(title)
}
get_content = function(inside_link){
  inside_page = read_html(inside_link)
  content = inside_page %>%  html_nodes("p") %>% html_text() %>% paste(collapse = " ")
  return(content)
}
get_date = function(inside_link){
  inside_page = read_html(inside_link)
  date = inside_page %>%  html_nodes(".published_time") %>% html_text() %>% sub("^Publish\\s*:\\s*", "", .)
  return(date)
}
```

Output:

Functions return respective article title, content, and published date.

1.1.4 Define scrape_category() Function

Description:

The scrape_category() function automates the scraping of news articles from a given category page on the Dhaka Tribune website. It first navigates to the specified URL using RSelenium, waits for the page to load, and calls clickAjaxButton() to reveal more articles. Then, it extracts the article links using CSS selectors and applies the previously defined scraping functions (get_title, get_content, get_date) to each link. The result is compiled into a data frame with columns for title, date, content, and category. This function encapsulates the full workflow for scraping a single news section.

Code:

```r
scrape_category <- function(section_url, category, max_articles = 100) {

  remote_driver$navigate(section_url)
  Sys.sleep(5)

  clickAjaxButton(times = 10, button_selector = "#ajax_load_more_704_btn")

  page_source <- remote_driver$getPageSource()[[1]]
  page <- read_html(page_source)
  inside_link <- page %>% html_nodes(".link_overlay") %>% html_attr("href") %>% paste("https:", ., sep = "") %>% head(max_articles)

  titles <- sapply(inside_link, FUN = get_title)
  dates <- sapply(inside_link, FUN = get_date)
  contents <- sapply(inside_link, FUN = get_content)

  df <- data.frame(
    title = titles,
    date = dates,
    content = contents,
    category = category,
    stringsAsFactors = FALSE
  )
  return(df)
```

Output:

Returns a data frame with title, date, content, and category columns.

1.1.5 Launch Selenium Firefox Driver

Description:

This code block prepares the environment to run RSelenium using the Firefox browser. First, it uses binman::list_versions("geckodriver") to list available versions of the GeckoDriver, which is needed to control Firefox. Then, netstat::free_port() finds an available network port to avoid conflicts when running the Selenium server. Using these, the rsDriver() function launches a Selenium server with Firefox configured to use GeckoDriver version 0.35.0 on the free port. The check = FALSE argument skips checking for driver updates to speed up initialization, and verbose = TRUE enables detailed logging. Finally, the Firefox remote driver client is extracted from the server object and assigned to remote_driver, which can be used to control the browser in subsequent commands.

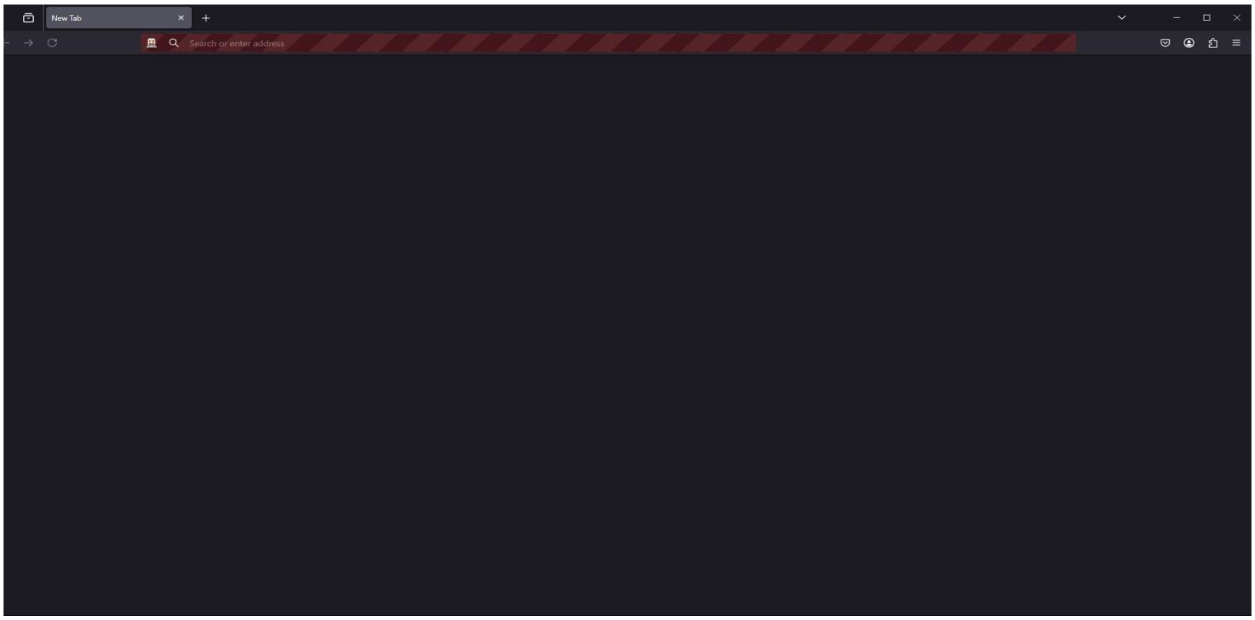
Code:

```r
binman::list_versions("geckodriver")
port <- netstat::free_port()
driver <<- rsDriver(browser = "firefox",
                    geckover="0.35.0",
                    chromever=NULL,
                    check = FALSE,
                    port = port,
                    verbose = TRUE
)

remote_driver <- driver[["client"]]
```

Output:

Firefox browser opens and becomes ready for automated interaction.



1.1.6 Scrape Multiple Categories

Description:

Using the scrape_category() function, this section extracts news data from five different categories on the Dhaka Tribune website: sports, politics, entertainment, foreign affairs, and elections. Each category is accessed by its respective URL, and up to 100 articles are scraped per category. The scraped data from each section is stored in separate data frames to preserve category-wise separation before being combined later.

Code:

```r
sports_data <- scrape_category("https://www.dhakatribune.com/sport", "sports")
politics_data <- scrape_category("https://www.dhakatribune.com/bangladesh/politics", "politics")
entertainment_data <- scrape_category("https://www.dhakatribune.com/showtime", "entertainment")
foreign_affairs_data <- scrape_category("https://www.dhakatribune.com/bangladesh/foreign-affairs", "foreign_affairs")
election_data <- scrape_category("https://www.dhakatribune.com/bangladesh/election", "election")
```

Output:

Five data frames with up to 100 articles each. The sports data frame is shown below:

	title	date	content	category
https://www.dhakatribune.com/sport/tennis/381440/sinner-...	Sinner lines up Alcaraz showdown in Italian Open final	17 May 2025, 08:13 AM	Alcaraz booked his place in the showdown by beating Sinne...	sports
https://www.dhakatribune.com/sport/football/381439/villa-...	Villa sink Spurs to bolster Champions League bid	17 May 2025, 08:06 AM	2nd-half goals from Konsa and Kamara ensured Villa's battl...	sports
https://www.dhakatribune.com/sport/football/381438/slot-s...	Slot's mind on next season but tight-lipped on Frimpong pu...	17 May 2025, 07:56 AM	Liverpool are reportedly closing in on a deal for Frimpong a...	sports
https://www.dhakatribune.com/sport/football/381437/artet...	Arteta ready to smash transfer budget to sign striker	17 May 2025, 07:51 AM	Arteta's side have now gone 5 years without a trophy as the...	sports
https://www.dhakatribune.com/sport/cricket/381435/bangl...	Bangladesh storm into SAFF U-19 Championship final	16 May 2025, 11:52 PM	Defending champions Bangladesh stormed into the final of ...	sports
https://www.dhakatribune.com/sport/cricket/381406/starc-...	Starc opts out of return to IPL	16 May 2025, 02:59 PM	Starc told the Capitals he won't return, freeing him up to foc...	sports
https://www.dhakatribune.com/sport/football/381392/how-...	How Flick's Barca wrestled LaLiga back from Real Madrid	16 May 2025, 09:24 AM	Barca clinched the Spanish title Thursday for the 28th time i...	sports
https://www.dhakatribune.com/sport/formula-one/381388/...	Hamilton admits he underestimated Ferrari challenge	16 May 2025, 09:02 AM	After 6 races this year, he is still seeking a 1st podium finish ...	sports
https://www.dhakatribune.com/sport/football/381387/de-br...	De Bruyne seeks fitting City farewell in FA Cup final	16 May 2025, 08:56 AM	A 6-time Premier League winner, De Bruyne could add a 3rd...	sports
https://www.dhakatribune.com/sport/cricket/381386/west-i...	West Indies issues LA 2028 Olympic cricket plea	16 May 2025, 07:48 AM	LA 2028 will see cricket returning to the Olympics for the 1s...	sports
https://www.dhakatribune.com/sport/football/381385/rashf...	Rashford to miss final two games of Villa's season	16 May 2025, 07:42 AM	There are reports Rashford wants to play Champions Leagu...	sports
https://www.dhakatribune.com/sport/football/381384/amor...	Amorim urges Man Utd to focus on Chelsea, not Europa final	16 May 2025, 07:27 AM	Dalot, De Ligt and Yoro are all expected to miss the trip to L...	sports
https://www.dhakatribune.com/sport/cricket/381383/stokes...	Stokes determined to dominate on England return	16 May 2025, 07:22 AM	The 33-year old is set to return in a 1-off Test against Zimba...	sports
https://www.dhakatribune.com/sport/football/381382/haala...	Haaland eyes FA Cup to save face after City's horrific season	16 May 2025, 07:16 AM	Haaland said City must deliver in the FA Cup final Saturday t...	sports
https://www.dhakatribune.com/sport/cricket/381381/anders...	Anderson set for Lancashire return	16 May 2025, 06:06 AM	He has been included in Lancashire's squad for their match ...	sports
https://www.dhakatribune.com/sport/football/381380/alba-...	Alba inks Inter Miami extension to 2027	16 May 2025, 05:59 AM	The 36-year old defender, who joined Miami in 2023 after a...	sports

Showing 1 to 17 of 100 entries, 4 total columns

1.1.7 Combine and Save Raw Data

Description:

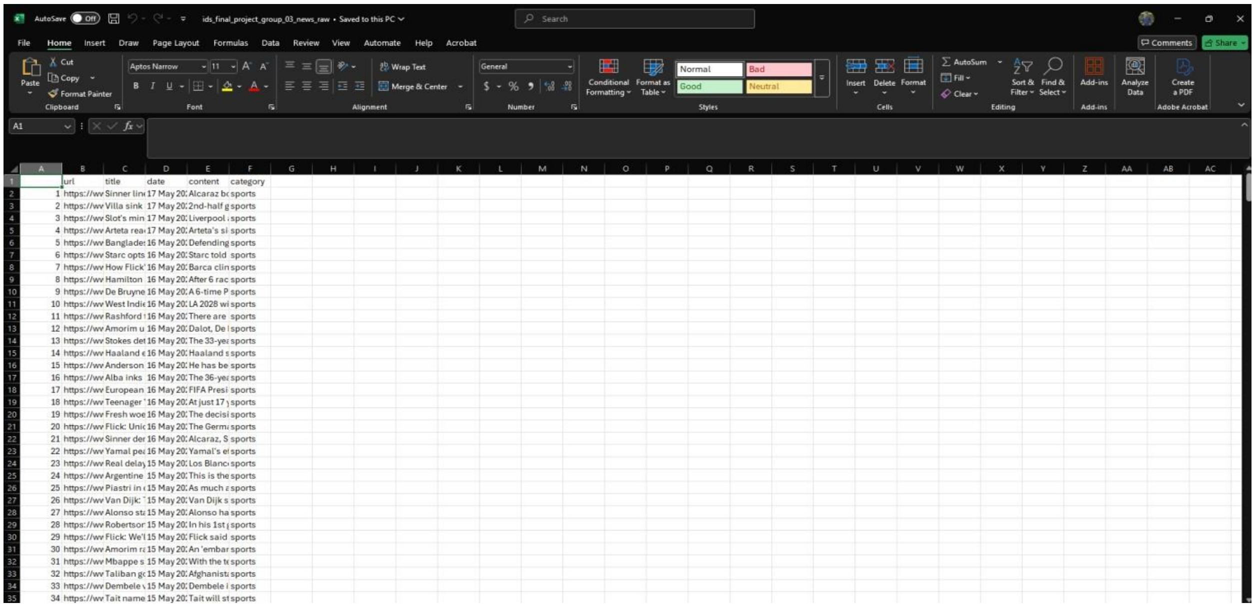
Once all categories have been scraped, the individual data frames are merged into one consolidated data frame using bind_rows() from the dplyr package. This combined dataset contains all the scraped news articles across various topics. The final data frame is then written to a CSV file using write.csv(), storing the raw data on disk for further analysis or processing.

Code:

```
df <- bind_rows(sports_data, politics_data, entertainment_data, foreign_affairs_data, election_data)
write.csv(df, "D:/Study Materials/SPRING_2024-2025/data science/codes/ids_final_project_group_03_news_raw.csv")
```

Output:

CSV file saved containing all raw news articles.



1.1.8 Close Browser and Selenium Server

Description:

After the scraping process is complete, it is important to properly terminate the browser and Selenium server to free up system resources. This is done by calling remote_driver$close() to close the browser and driver$server$stop() to shut down the local Selenium server. Ensuring a clean shutdown helps avoid memory leaks or port conflicts in future sessions.

Code:

```

remote_driver$close()
driver$server$stop()

```

Output:

Firefox browser closes and server shuts down.

PART 1.2: TEXT PREPROCESSING

1.2.1 Load Required Libraries

Description:

Before text preprocessing begins, several additional libraries are loaded. The stopwords package provides standard lists of stopwords in various languages. textclean offers tools to clean text data, including removing emojis and expanding contractions. textstem provides lemmatization functionality to reduce words to their base forms. Lastly, stringr is used for advanced string manipulation.

Code:

```
library(dplyr)
library(stopwords)
library(textclean)
library(textstem)
library(stringr)
```

Output:

Libraries are loaded for text processing.

1.2.2 Define Preprocessing Functions

Description:

Several preprocessing functions are defined to clean and standardize the raw text data. clean_text() lowercases the text, removes punctuation, and trims excess whitespace. tokenization() splits text into individual words. remove_stopwords() filters out common English stopwords using the stopwords package. remove_numeric_tokens() eliminates numeric tokens that are generally irrelevant for text analysis. These are all combined into preprocess_pipeline(), which also applies replace_emoji() and replace_contraction() from textclean, and lemmatize_words() from textstem, returning a cleaned and normalized version of the input text.

Code:

```r
tokenization <- function(text_vector) {
  strsplit(text_vector, " ")
}

remove_stopwords <- function(token_list) {
  stop_words <- stopwords("en")
  lapply(token_list, function(words) {
    words[!(words %in% stop_words)]
  })
}

remove_numeric_tokens <- function(token_list) {
  lapply(token_list, function(words) {
    words[!grepl("[0-9]", words)]
  })
}
```

```r
preprocess_pipeline <- function(corpus) {
  corpus <- replace_emoji(corpus)
  corpus <- replace_contraction(corpus)
  corpus <- clean_text(corpus)
  tokens <- tokenization(corpus)
  tokens <- remove_stopwords(tokens)
  tokens <- lapply(tokens, lemmatize_words)
  tokens <- remove_numeric_tokens(tokens)
  combined_text <- sapply(tokens, paste, collapse = " ")
  return(combined_text)
}
```

Output:

Returns a cleaned and preprocessed version of input text. An example is given below:

```
> preprocess_pipeline("I can't wait to test this! 😊 Running quickly in 2024.")
[1] "can wait test smile face smile eye run quickly"
>
```

1.2.3 Preprocess and Save Cleaned Text

Description:

In the final step, the raw data is read from the previously saved CSV file using read.csv(). The preprocess_pipeline() function is then applied to the content column, which holds the article bodies. The resulting cleaned text is stored in a new data frame named processed_df and written to a separate CSV file using write.csv(). This preprocessed data is now ready for topic modeling.

Code:

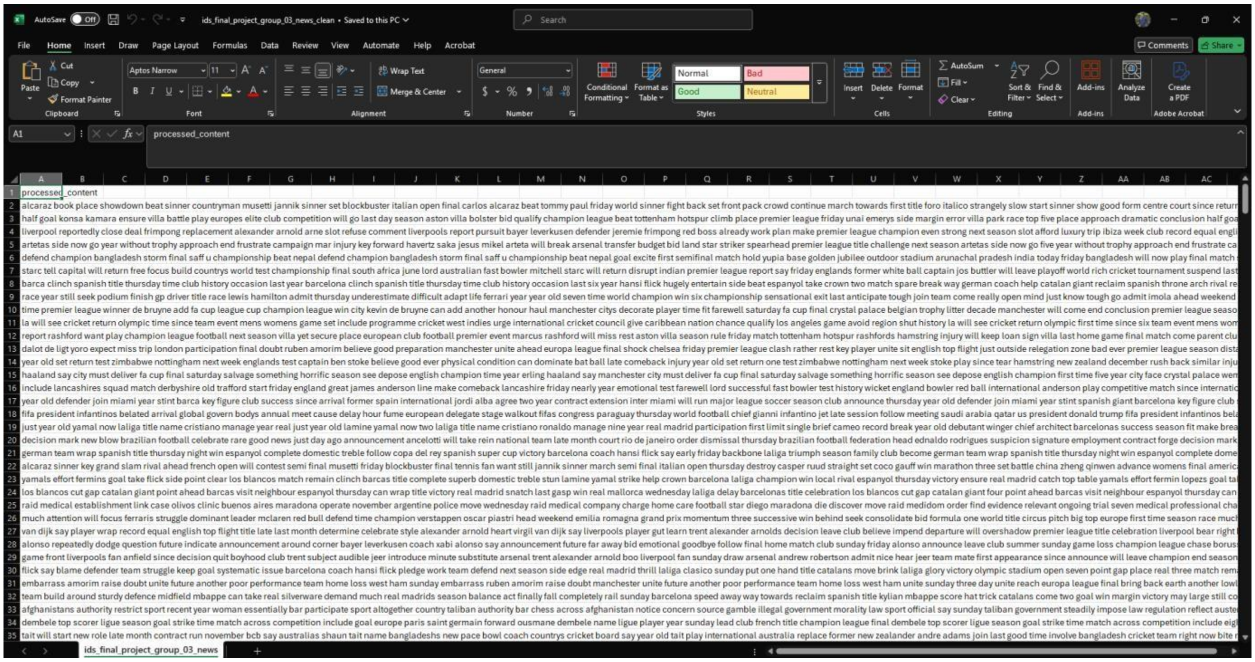
```r
df <- read.csv("D:/Study Materials/SPRING_2024-2025/data science/codes/ids_final_project_group_03_news_raw.csv", stringsAsFactors = FALSE)

processed_content <- preprocess_pipeline(df$content)
processed_df <- data.frame(processed_content = processed_content)

write.csv(processed_df, "D:/Study Materials/SPRING_2024-2025/data science/codes/ids_final_project_group_03_news_clean.csv", row.names = FALS
```

Output:

Cleaned text content is saved in a new CSV.


ids_final_project_group_03_news_clean • Saved to this PC
processed_content
ids_final_project_group_03_news

PART 2: TOPIC MODELING USING LDA

2.1 Load Required Libraries

Description:

This segment loads all the necessary libraries for topic modeling. The dplyr package is used for efficient data manipulation, such as grouping and summarizing the top terms per topic. The tm (Text Mining) package handles the creation and processing of a text corpus, particularly for generating a Document-Term Matrix (DTM) from text data. The topicmodels package enables Latent Dirichlet Allocation (LDA) to extract topics from text. tidytext allows the output from LDA to be converted into tidy data frames that integrate well with dplyr and other tidyverse tools. reshape2 and tidyr are both used for reshaping and cleaning data structures, such as converting long tables of topic terms into wide format. Lastly, knitr is used to neatly format the results into readable tables, particularly useful when presenting the top terms per topic.

Code:

```r
library(dplyr)
library(tm)
library(topicmodels)
library(tidytext)
library(ggplot2)
library(reshape2)
library(knitr)
library(tidyr)
```

Output:

All libraries are loaded with no error.

2.2 Data Loading and Preprocessing

Description:

The cleaned dataset is loaded from a CSV file using read.csv() and then transformed into a corpus using VCorpus() and VectorSource() functions from the tm package. This step prepares the text data for further analysis by organizing it into a structured format.

Code:

```r

df <- read.csv("D:/Study Materials/AIUB/SPRING_2024-2025/data science/codes/ids_final_project_group_03_news_clean.csv", stringsAsFactors = FA

corpus <- VCorpus(VectorSource(df$processed_content))

```

Output:

Example of the 1st text document after transformed into a corpus:

```
> inspect(corpus[[1]])
<<PlainTextDocument>>
Metadata:  7
Content:  chars: 2275

alcaraz book place showdown beat sinner countryman musetti jannik sinner set blockbuster italian open final carlos alcara
z beat tommy paul friday world sinner fight back set front pack crowd continue march towards first title foro italico str
angely slow start sinner show good form centre court since return action last week three month dope ban take unbeaten run
match year old will face alcaraz last man beat sinner final china open early october eye another potential final pair fre
nch open next month win sinner rival see mens rome title go italian first time since adriano panatta want win sunday play
one good tennis sure say sinner carlos play incredible tennis let us see come side know incredible final alcaraz book pla
ce showdown beat sinner countryman lorenzo musetti four time grand slam champion overcome musetti windy condition just tw
o hour reach final season go dinner phone go watch sinner match say alcaraz win know go play watch match see go play muse
tti beat alcaraz monte carlo final last month fall straight defeat spaniard frustrate display believe alcaraz will good s
inner bring top form sunday even really rate carlos think clay good version carlos favourite anyonethat include jannik te
ll reporter paul rattle first five game minute near replica sinner casper ruud thursday close first set little half hour
last time sinner lose set quarter final us open daniil medvedev match win way grand slam triumph sinner look shadow playe
r dominate tennis throughout right start suspension agree world anti dope agency early february nowhere come roar back se
t finally force paul back deep baseline shot first ace match win set love level match complete role reversal paul now one
throw around court world win just point set look bewilder quickly momentum shift paul hand sinner initiative double fault
night game two set italian eventually win nine game row march victory early jasmine paolini continue bid win womens singl
e double title rome reach final week time alongside fellow italian sara errani paolini errani reign double champion will
meet veronika kudermetova elise mertens final sunday paolini already crow single champion late bloomer paolini take coco
gauff saturday aim series title age become first italian woman win rome since raffaella reggi
```

2.3 Document-Term Matrix Creation

Description:

This code creates a Document-Term Matrix (DTM) from the cleaned text corpus using the DocumentTermMatrix() function. The control parameter is set to wordLengths = c(3, Inf) to include only words with three or more characters, filtering out short, less meaningful terms. Each row in the matrix represents a document, and each column represents a unique term, with values indicating word frequency. dim(dtm) prints the size of the matrix, and as.matrix(dtm)[1:5, 1:10] shows a preview of the first 5 documents and 10 terms, illustrating the sparse nature of the data.

Code:

```
dtm <- DocumentTermMatrix(corpus, control = list(wordLengths = c(3, Inf)))
print(dim(dtm))
print(as.matrix(dtm)[1:5, 1:10])
```

Output:

```
> dtm <- DocumentTermMatrix(corpus, control = list(wordLengths = c(3, Inf)))
> print(dim(dtm))
[1]   500 10190
> print(as.matrix(dtm)[1:5, 1:10])
    Terms
Docs aadhaar aagey aaj aaqib abandon abbas abcha abdominal abduct abduction
   1       0     0   0     0       0     0     0         0      0         0
   2       0     0   0     0       0     0     0         0      0         0
   3       0     0   0     0       0     0     0         0      0         0
   4       0     0   0     0       0     0     0         0      0         0
   5       0     0   0     0       0     0     0         0      0         0
>
```

2.4 Topic Modeling with LDA

Description:

This line applies Latent Dirichlet Allocation (LDA) to the Document-Term Matrix using the LDA() function from the topicmodels package. The argument k = 10 sets the number of topics the model will attempt to discover. The control = list(seed = 1234) ensures reproducibility by setting a fixed random seed. The model generates two key probability distributions that help reveal the underlying themes: Beta (β) and Gamma (γ). B is the probability of each word given a topic (Topic-to-Term distribution) and γ is the proportion of each topic within a document (Document-to-Topic distribution. β values indicate how strongly a word is linked to a specific topic, while γ values show how much a topic contributes to an individual document.

Code:

```r

lda_model <- LDA(dtm, k = 10, control = list(seed = 1234))

```

Output:

```
> lda_model
A LDA_VEM topic model with 10 topics.
>
```

2.5 Extracting Top Terms per Topic

Description:

In this step, the most representative terms for each topic are extracted to aid in interpreting the LDA model. The tidy() function from the tidytext package is used with the argument matrix = "beta" to convert the topic-term probabilities into a tidy data frame format. Then, the code groups the data by topic and selects the top 10 terms with the highest beta values using group_by() and top_n(), which indicate how strongly each word is associated with its respective topic. After ungrouping, the terms are sorted in descending order of probability within each topic using arrange(). Finally, summarise() and paste() are used to concatenate the top terms into a readable string for each topic, and the results are printed to the console.

Code:

```r
top_terms <- topics_terms %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

cat("Top 10 words per topic:\n")
top_terms %>%
  group_by(topic) %>%
  summarise(top_words = paste(term, collapse = ", ")) %>%
  arrange(topic) %>%
  print(n = 10)
```

Output:

```
# A tibble: 10 × 2
   topic top_words
   <int> <chr>
 1     1 bangladesh, will, song, music, pope, chief, pilgrim, perform, good, hajj
 2     2 film, say, make, festival, director, story, one, time, work, also
 3     3 league, win, season, final, say, good, champion, year, will, team
 4     4 election, say, commission, will, voter, national, reform, commissioner, government, meet
 5     5 vote, party, also, expatriate, system, voter, among, political, method, ballot
 6     6 bnp, will, say, rahman, khaleda, return, zia, leader, dal, tarique
 7     7 say, will, worker, india, pakistan, indian, cricket, player, also, team
 8     8 will, test, film, day, year, star, series, khan, movie, new
 9     9 bangladesh, say, adviser, will, chief, meet, also, country, support, foreign
10    10 say, party, government, bnp, election, awami, will, political, league, people
>
```

Result Interpretation:

The topic modeling results identify distinct themes across the dataset. Topic 1 revolves around Bangladesh, song, pope, and hajj, suggesting a focus on religious, cultural, possibly representing foreign affairs. Topic 2, with words like film, director, and story, clearly reflects the entertainment industry. Topic 3, including league, champion, and season, is centered on sports, particularly team competitions. Topic 4 highlights election, commission, and reform, indicating electoral processes and governance. Topic 5 discusses votes, expatriate, and ballot, pointing to voting systems and political participation, especially for expatriates. Topic 6, mentioning BNP, Khaleda, and Tarique, is focused on BNP party politics. Topic 7, with India, Pakistan, and cricket, likely relates to international cricket, blending sports and foreign affairs. Topic 8, listing film, Khan, and series, again touches on entertainment, particularly celebrity media. Topic 9, featuring Bangladesh, foreign, and support, points to diplomatic relations and international meetings. Finally, Topic 10, including Awami, election, and government, deals with mainstream political discourse, involving major parties like Awami League and BNP.

2.6 Top Terms per Topic with Probability

Description:

This code ranks and formats the top 10 terms per topic based on their beta probabilities. It arranges terms by topic and descending beta values, selects the top 10 using slice_max(), and adds a rank and formatted term-probability string. The output is reshaped into a wide Table with pivot_wider(), where each column represents a topic. Finally, kable() is used to display a clean table titled “Top 10 Terms per Topic with Probability.”

Code:

```r
top_terms_wide <- top_terms %>%
  arrange(topic, desc(beta)) %>%
  group_by(topic) %>%
  slice_max(beta, n = 10) %>%
  mutate(Rank = row_number(),
         Term_with_Prob = paste0(term, " (", round(beta, 4), ")")) %>%
  select(topic, Rank, Term_with_Prob) %>%
  pivot_wider(names_from = topic, values_from = Term_with_Prob, names_prefix = "Topic_")
kable(top_terms_wide, caption = "Top 10 Terms per Topic with probability")
```

Output:

Table: Top 10 Terms per Topic with probability

Rank	Topic_1	Topic_2	Topic_3	Topic_4	Topic_5	Topic_6	Topic_7	Topic_8	Topic_9	Topic_10
1	bangladesh (0.0165)	film (0.029)	league (0.0138)	election (0.0613)	vote (0.024)	bnp (0.0189)	say (0.0133)	will (0.0153)	bangladesh (0.0271)	say (0.0259)
2	will (0.016)	say (0.0109)	win (0.0131)	say (0.031)	party (0.0103)	will (0.0149)	will (0.0122)	test (0.0134)	say (0.0202)	party (0.0216)
3	song (0.0095)	make (0.0074)	season (0.0117)	commission (0.0292)	also (0.0087)	say (0.0137)	worker (0.0101)	film (0.0106)	adviser (0.0143)	government (0.0174)
4	music (0.0072)	festival (0.0068)	final (0.0114)	will (0.0185)	expatriate (0.0075)	rahman (0.0129)	india (0.0087)	day (0.0082)	will (0.0098)	bnp (0.0149)
5	pope (0.0068)	director (0.0068)	say (0.0101)	voter (0.0124)	system (0.0065)	khaleda (0.0117)	pakistan (0.0086)	year (0.0079)	chief (0.0089)	election (0.0141)
6	chief (0.0067)	story (0.0065)	good (0.0093)	national (0.0123)	voter (0.006)	return (0.0113)	indian (0.0074)	star (0.0077)	meet (0.008)	awami (0.0133)
7	pilgrim (0.0065)	one (0.0065)	champion (0.0088)	reform (0.0113)	among (0.0056)	zia (0.0085)	cricket (0.0073)	series (0.0074)	also (0.0076)	will (0.0121)
8	perform (0.0064)	time (0.0064)	year (0.0086)	commissioner (0.011)	political (0.0046)	leader (0.0082)	player (0.0064)	khan (0.0071)	country (0.0076)	political (0.0117)
9	good (0.0055)	work (0.0058)	will (0.0083)	government (0.0097)	method (0.0046)	dal (0.0081)	also (0.0058)	movie (0.0068)	support (0.0074)	league (0.0113)
10	hajj (0.0054)	also (0.0054)	team (0.0079)	meet (0.0095)	ballot (0.0045)	tarique (0.0079)	team (0.0055)	new (0.0067)	foreign (0.0071)	people (0.0101)

Result Interpretation:

The inclusion of term-probability values reveals the strength of association between specific words and their corresponding topics, which is crucial for understanding thematic clarity and overlap.

In Topic 4, the word election (0.0613) has a notably higher probability than the other terms, such as say (0.031) and commission (0.0292). This sharp concentration indicates a focused theme—centered explicitly on electoral processes, reforms, and political events. In contrast, Topic 1 displays more thematic breadth. While bangladesh (0.0165) and song (0.0095) are among the top terms, their relatively close and moderate beta values suggest the topic blends cultural, national, and possibly religious content without a single dominating focus. Similarly, Topic 2 features film (0.029), say (0.0109), and make (0.0074). The high value of film indicates a strong entertainment or cinematic theme, but the inclusion of more general terms like say and make implies the topic may touch on broader aspects of media or narrative. Topic 10 also offers a blend of political discourse, with top terms such as say (0.0259), party (0.0216), and government (0.0174). The distribution here suggests political commentary or inter-party discussions, rather than a single event like an election. Meanwhile, Topic 5, with terms like vote (0.024), party (0.0103), and expatriate (0.0075), highlights diaspora political engagement, pointing to the participation of expatriates in elections. The relatively sharp decline in probabilities after the top term reflects a moderate thematic concentration.

By examining the strength and distribution of probabilities, we gain insights into:

- Which topics are narrow and focused (e.g., Topic 4),
- Which are broad and blended (e.g., Topic 1, Topic 10),
- And which may have distinct subthemes embedded within them (e.g., Topic 2, Topic 5).

2.7 Topic proportions for each document

Description:

Each document is associated with several topics, and their importance is determined by the γ (gamma) value. A high γ (close to 1) means the topic dominates that document, while a low γ indicates minimal relevance. This code extracts document-topic probabilities (gamma) from the LDA model using tidy(). It selects two random documents with sample() to demonstrate how topics are distributed across articles. The filter() and arrange() functions are used to display the proportion of each topic within those documents. Finally, print(n = 10) shows the topic proportions for the selected documents, revealing which themes dominate their content.

Code:

```r

doc_topics <- tidy(lda_model, matrix = "gamma")
set.seed(123)
random_docs <- sample(unique(doc_topics$document), 2)
cat("\nTopic proportions for 10 random documents:\n")
doc_topics %>%
  filter(document %in% random_docs) %>%
  arrange(document, topic) %>%
  print(n = 10)

```

Output:

```
# A tibble: 20 × 3
   document topic    gamma
   <chr>    <int>    <dbl>
 1 415          1 0.000131
 2 415          2 0.000131
 3 415          3 0.000131
 4 415          4 0.576
 5 415          5 0.000131
 6 415          6 0.000131
 7 415          7 0.000131
 8 415          8 0.000131
 9 415          9 0.000131
10 415         10 0.423
# ℹ 10 more rows
# ℹ Use `print(n = ...)` to see more rows
```

Result Interpretation:

In case of Document 415:

- Topic 4 (Election): $\gamma = 0.576 \rightarrow$ Primary Theme
- Topic 10 (Awami League Politics): $\gamma = 0.423 \rightarrow$ Secondary Theme
- Other Topics: $\gamma \approx 0.0001 \rightarrow$ Not significant

Document 415 primarily discusses elections, commission roles, and political governance involving the Awami League. The high γ values for Topics 4 and 10 confirm its strong political and electoral context. Other topics are nearly absent. Clearly indicating a mixture between election and politics category.