Strings

Text Mining

Overview

- Text – third type of data.
  - Unstructured
  - Not as direct to analyze compared to structured data
  - Subjective Interpretation
- Objective: To automate processing and analysis of text. i.e. Software/Machine to “read” text, not humans.
- Part 1: Processing Strings
  - Base R
  - stringr package
- Part 2: Text Mining and Sentiment Analysis
  - quanteda package

What is a String?

- A string is a sequence of characters:
  - “hey man”
  - “12 Apr”
  - “#$^ 5 7 d g”
  - “ “
  - 5 vs “5”
- In R, this is an object of class character (chr).
- Many functions available to define and manipulate strings.

Part 1: Processing Strings

Based on Main Textbook Chapter 10: Strings and Text Mining

String Representation

- What's the difference between:
  1. Who
  2. "Who"

Ans:

1. Who is the name of an object and could be a constant, vector, data.frame,…, etc. e.g. Who <- 2

2. "Who" is a text string.

Counting Characters

- nchar() counts the number of characters in a string.
- length() counts the number of elements in a vector.

```
# How many characters in 3 strings?
nchar(c("who", "are", "you?"))

## [1] 3 3 4

# How many characters in 1 string?
nchar("Who are you?")
## [1] 12
```

Includes white space & punctuation.

```
# How many elements in the vector?
length(c("who", "are", "you?"))

## [1] 3

length("Who are you?")

## [1] 1
```

Find and Replace character(s)

```r
# Character Translation with chartr(old, new, x) function.
# replace a by A
chartr(old = "a", new = "A", "This is a very interesting seminar.")

## [1] "This is A very interesting seminAr."

# strings Old and New must have same length.
# a replaced by X, i replaced by Z.
chartr("ai", "XZ", "This is a very interesting seminar.")

## [1] "ThZs Zs X very ZnterestZng semZnXr."
```

Extract Substring with substr(x, start, stop)

```
# slice and extract substring from 2nd position to 4th position
substr("abcde", start = 2, stop = 4)

## [1] "bcd"

# replace 2nd character with # symbol
x <- c("Today", "is", "a", "hot", "day")
substr(x, start = 2, stop = 2) <- "#"
x

## [1] "T#day" "i#"    "a"     "h#t"   "d#y"

# replace 2nd and 3rd characters with happy face symbol
y <- c("Today", "is", "a", "hot", "day")
substr(y, start = 2, stop = 3) <- ":)"
y

## [1] "T:)ay" "i:"    "a"     "h:)"   "d:)"
```

Difference between chartr() and substr()

```
chartr(old = "a", new = "A", "This is a very interesting seminar.")

## [1] "This is A very interesting seminAr."
```

```
x <- c("Today", "is", "a", "hot", "day")
substr(x, start = 2, stop = 2) <- "#"
x

## [1] "T#day" "i#"    "a"     "h#t"   "d#y"
```

Q: What's the difference between chartr() and substr() approach to replace character in a string?

chartr() requires a specific character to be replaced while substr() requires the position in the string to be replaced.

Changing Case

```r
# To upper case in a vector with 2 strings.
toupper(c("All in Upper Case.", "abcde"))

## [1] "ALL IN UPPER CASE." "ABCDE"

# to lower case in a vector with 2 strings.
tolower(c("ALL in Lower Case.", "ABCDE"))

## [1] "all in lower case." "abcde"
```

R package *stringr*

- Extends the string processing capabilities in base R
- Provide more consistent syntax for many string related functions – str_*
  - i.e. Many (but not all) functions within stringr package starts with str_
- Need to install stringr package.

Counting Characters with str_length()

```
library(stringr)
# How many characters in 3 strings?
str_length(c("who", "are", "you?"))

## [1] 3 3 4

# How many characters in 1 string?
str_length("Who are you?")

## [1] 12
```

Same results as nchar() from base R.

Combine strings into one string with *str_c()*

```
x <- "news"
str_c("This", "is", "interesting", x)

## [1] "Thisisinterestingnews"

# Separates each string with a white space.
str_c("This", "is", "interesting", x, sep =" ")

## [1] "This is interesting news"
```

str_sub() function is similar to *substr()* in base R

- But allows negative integers for start and end.
- Negative start or end look from the right end of the string.

```
str_sub("abcde", start = 2, end = 4)

## [1] "bcd"

# Get the third last and second last characters using negative start and e
nd
str_sub("abcde", start = -3, end = -2)

## [1] "cd"
```

Extract word(s) from string(s) with *word()*

```
sw <- "Snow White and the 7 strong men"

# Extract Words with word(string, start = 1, end = start, sep = fixed(" ")
)

# extract first word
word(sw, start = 1)

## [1] "Snow"

# extract first 2 words
word(sw, start = 1, end = 2)

## [1] "Snow White"
```

```
# extract last 2 words
word(sw, start = -2, end = -1)

## [1] "strong men"
```

Reference for String Processing

- Sanchez (2018). Handling Strings with R. eBook: https://www.gastonsanchez.com/r4strings/

Test your understanding of String Computations

Complete Exercise 10.1 Q1 – Q6.

Text Mining

Text Mining

String Processing vs Text Mining

“ For basic manipulation of strings, base R and/or stringr package would be sufficient.

However, for longer text e.g. speeches, documents, reviews, news, books, etc… or deeper analysis of text e.g. sentiments, keywords, choice of words, …etc, a text mining package is useful and will tremendously boost text analysis productivity.

There are at least two popular text mining packages in R – (a) tidytext and (b) quanteda. We will focus on quanteda R package in this chapter.”

--- Chew C.H. (2020) AAD1 Chap 10

Part 2: Text Mining & Sentiment Analysis

Based on AAD1 Textbook Chapter 10: Strings and Text Mining

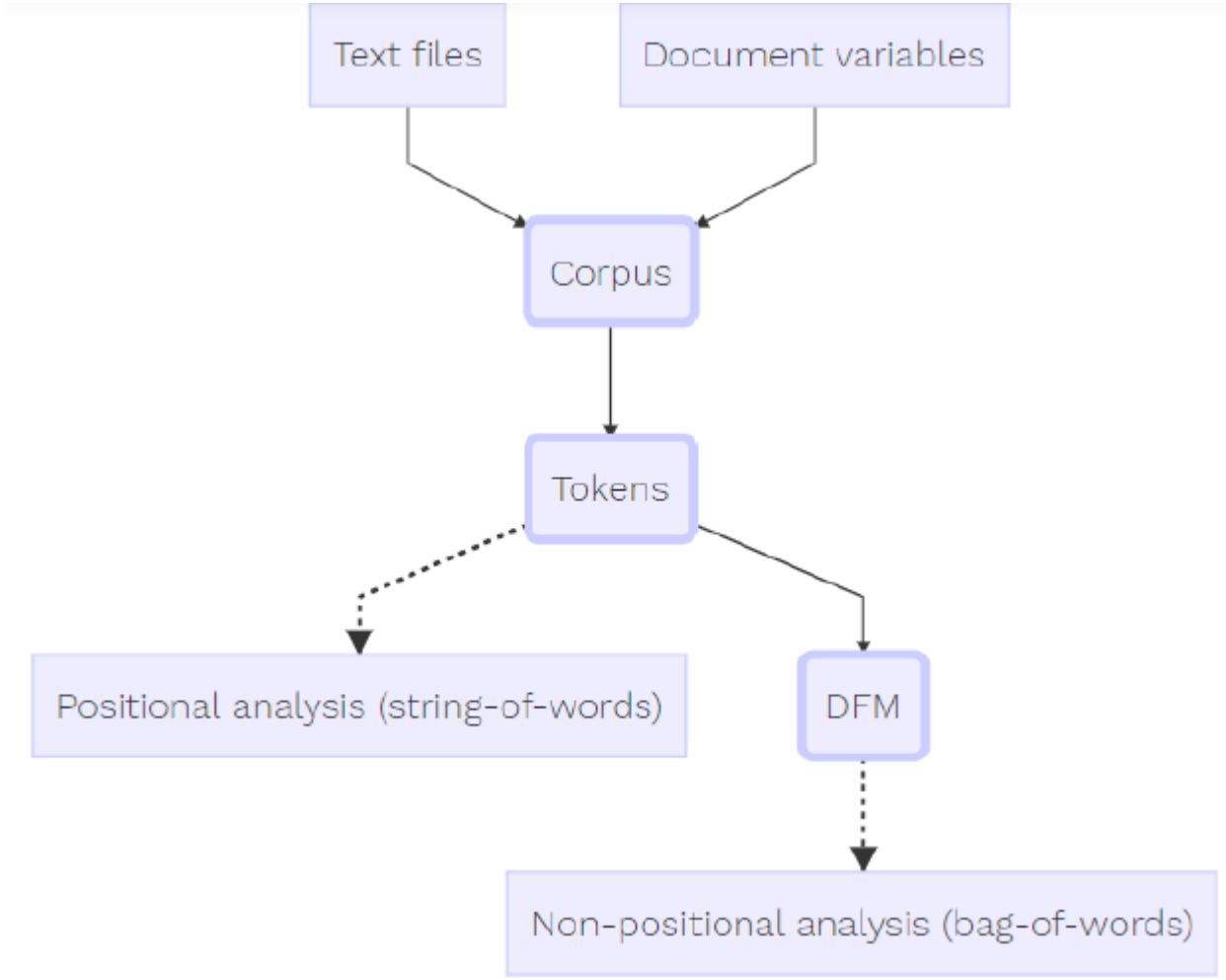
Basic Concepts in Text Mining

1. Corpus
2. Tokens
3. DFM (Document Feature Matrix), aka Document Term Matrix.
4. Stopwords
5. Stemming
6. Dictionary
7. Keywords in Context (KWIC)

Corpus and Tokens

- Corpus
  - Collection of textual content in all the text documents.
  - Metadata about each text document.
  - CSV or Excel: One document per row
  - Folder of PDFs, word docs, text files, etc.
- Tokens
  - The unit of text analysis that will be performed on the Corpus.
  - Default: Single Word.
- DFM
  - Row: One Document
  - Column: One token
  - Cell: Frequency count of that token in that document.

Relationship between Corpus, Tokens and DFM.



Source: Chew C. H. (2019) Analytics, Data Science and AI Vol. 1 Chap 10, Figure 10.1

Stopwords

- Stopwords
  - A list of words to be removed

There are eighteen lists of stopwords in various languages within *quanteda* package. For the full list, their sources and examples, see https://quanteda.io/reference/stopwords.html

```
# first 6 stopwords in the English stopword list.
head(stopwords("en"))

## [1] "i"      "me"     "my"     "myself" "we"     "our"

# first 6 stopwords in the Chinese stopword list.
head(stopwords("chinese"))

## [1] "按"   "按照" "俺"   "俺"   "们"   "阿"
```

Stemming

- Different words can provide the same meaning or informational content.
- Stemming reduces all words to their basic stem and thus, treat all such different expressions as the same “word”.

```
char_wordstem(c("run", "running", "runs", "runner", "Run"))

## [1] "run"    "run"    "run"    "runner" "Run"
```

Dictionary (aka Lexicon)

- Match
  - Count
  - Translate Meaning (e.g. sentiment, emotion, etc.)

- Sentiment Analysis Dictionaries
  - Bing
  - NRC
  - Lexicoder

Bing Lexicon. Dataset: bing.csv

Bing lexicon is a collection of 6786 words, some deliberately misspelled as they are common spelling errors in social media, and tagged with either Positive or Negative sentiments.

```
bing <- read.csv("D:/Dropbox/Datasets/ADA1/9_TM/lexicons/bing.csv", string
sAsFactors = F)
bing[sample(nrow(bing), 15),]

##                word sentiment
## 535           bitch  negative
## 3755           leak  negative
## 634         braggart  negative
## 5335       sensation  positive
## 5354          severe  negative
## 3089        illusion  negative
## 4032            miff  negative
## 4173         mundane  negative
## 2917      hedonistic  negative
## 3398       infuriate  negative
## 4420 oversimplified  negative
## 15            abrupt  negative
## 6121      tragically  negative
## 1896    earsplitting  negative
## 2527         freezing  negative
```

NRC lexicon. Dataset: nrc.csv

NRC lexicon consists of 13,901 words and tagged with one of the following sentiments:

- Positive
- Negative
- Anger
- Anticipation
- Disgust
- Fear
- Joy
- Sadness
- Surprise
- Trust

	word	sentiment
230	adultery	sadness
7485	law	trust
7617	lie	disgust
362	ail	negative
6784	inefficient	negative
11870	stillborn	sadness
7	abandoned	negative
12542	toils	negative
201	admirable	trust
4922	favorable	surprise
13087	unpaid	negative

Lexicoder Sentiment Dictionary is in *quanteda* package

- 2858 negative words
- 1709 positive words
- 1721 negated positive words
- 2860 negated negative words.

```
##  Sample of 10 negative words in Lexicoder:

data_dictionary_LSD2015[[1]][sample(2858, 10)]
```

```
##  [1] "unhapp*"    "admonish*"  "foundered"  "virulent"   "madd*"
##  [6] "illusory"   "cast down*" "bully*"     "contraven*" "jeer*"
```

```
##  Sample of 10 positive words in Lexicoder:

data_dictionary_LSD2015[[2]][sample(1709, 10)]

##  [1] "sustain*"     "curious*"     "willingly*"   "supereminen*"
##  [5] "zest*"        "brainy"       "outliv*"      "snug*"
##  [9] "reassur*"     "consistent"
```

```
##  Sample of 10 negated positive words in Lexicoder:

data_dictionary_LSD2015[[3]][sample(1721, 10)]

##  [1] "not amatory*"      "not fun"           "not cheerful*"
##  [4] "not perfects"      "not love"          "not notori*"
##  [7] "not under control" "not strifeless*"   "not superwomen"
## [10] "not champion*"
```

```
##  Sample of 10 negated negative words in Lexicoder:

data_dictionary_LSD2015[[4]][sample(2860, 10)]

##  [1] "not jail*"          "not fractur*"       "not unwelcom*"
##  [4] "not whips"          "not vicious*"       "not homeli*"
##  [7] "not overcompensat*" "not roughed"        "not coerc*"
## [10] "not fault"
```

Note about Dictionaries

- The choice of words and association with sentiment or emotion is the perspective of the dictionary creator.
- It does not mean we must agree.
- Is “quiet” in classroom, a positive word?
- Feel free to amend that dictionary or use another dictionary.

Quanteda References for Beginners

https://quanteda.io/articles/quickstart.html

https://data.library.virginia.edu/a-beginners-guide-to-text-analysis-with-quanteda/

Test your understanding of Text Mining

Complete Exercise 10.1 Q7.

Summary

- Text Mining
  - Extraction of Information from text
  - After pre-processing to isolate useful information.
  - Automates the processing and analysis of text documents.
  - Output new input variables for predictive model.
  - Need to find a suitable dictionary.
  - Include human bias, stereotypes and conventions.
  - E.g. “miss” is negative sentiment in Bing lexicon. Miss Wong?
  - Faster and more “thorough” than human, but may or may not be more accurate than human.
  - Issues:
    - Sarcasm. E.g. He is such a good boy.
    - Domain specific words. E.g. DNA, CART, Burn rate.
    - Cultural differences. E.g. quiet student.