텍스트 마이닝(1)

숙명여자대학교 경영학부 오중산

텍스트 마이닝 소개

- 텍스트 마이닝 정의
  - 문자로 된 데이터에서 가치 있는 정보를 얻어 내는 분석 기법
  - 개인 온라인 활동이 확산됨에 따라 중요해진 분석 방법
- 텍스트 마이닝 단계별 구분
  - 단어 빈도 분석 / 형태소 분석기를 이용한 단어 빈도 분석 / 비교 분석
  - 감정 분석 / 의미망 분석 / 토픽 모델링

텍스트 전처리

- 텍스트 전처리란?
  - ◆ 텍스트에서 분석하는 데 불필요한 요소 제거
  - ◆ 텍스트를 다루기 쉬운 형태로 만드는 과정
- 텍스트 마이닝에서 활용할 자료
  - ◆ 문재인 대선 출마 선언문(text_moon.txt)

```
raw_moon <- readLines("speech_moon.txt", encoding = "UTF-8")
head(raw_moon)
```

텍스트 전처리

- 불필요한 문자 제거하기
  - ◆ 한글을 제외하고 모두 제거하기

```
txt <- "치킨은!! 맛있다. xyz 정말 맛있다!@#"
```

```
install.packages("stringr")
library(stringr)

str_replace_all(string = txt, pattern = "[^가-힣]", replacement = " ")
```

`string` : 처리할 텍스트

`pattern` : 규칙 `[^가-힣]` : 한글이 아닌 모든 문자

`replacement` : 바꿀 문자

텍스트 전처리

- raw_moon에서 불필요한 문자 제거하기

```
moon <- raw_moon %>%
 str_replace_all("[^가-힣]", " ")
```

- moon에서 연속된 공백 제거하기

```
moon <- moon %>%
 str_squish()
```

- 데이터 구조를 tibble로 바꾸기
  - ◆ 문장이 길면 보기 힘들기 때문에 문자열 벡터를 tibble 형태로 변경

```
library(dplyr)
moon <- as_tibble(moon)
```

텍스트 전처리

- 데이터 프레임과 비교한 tibble의 특성
  - ◆ tibble 형태 데이터를 실행하면 console 창에서 행과 열의 개수를 제시
  - ◆ 변수의 척도도 보여줌
  - ◆ 이후 텍스트를 토큰화하려면, tibble 형태로 저장해야 함
- 지금까지의 전처리 과정을 한줄 코드로 실현하기

```
moon <- raw_moon %>%
 str_replace_all("[^가-힣]", " ") %>%  # 한글만 남기기
 str_squish() %>%                      # 연속된 공백 제거
 as_tibble()                           # tibble로 변환
```

토큰화하기

- 토큰(token)이란?
  - ◆ 텍스트를 나눈 다양한 단위: 문장, 구절, 단어, 형태소 등
  - ◆ 토큰화란 텍스트를 토큰 형태로 만드는 것
- unnest_tokens 함수를 이용한 토큰화한 tibble 데이터 형성
  - ◆ tidytext 패키지에 있는 unnest_tokens 활용
    - ❖ tidytext 패키지는 dplyr, ggplot2와 함께 사용됨
    - ❖ input: tibble 형태 데이터에 있는 토큰화 대상 변수
    - ❖ output: 출력 변수명
    - ❖ token: 토큰 형태(sentences, words, characters)

```
word_space <- moon %>%
  unnest_tokens(input = value,
                output = word,
                token = "words")
```

단어 빈도 분석하기

- 단어 빈도 분석이란?
  - ◆ 어떤 단어가 얼마나 쓰였는지 분석함으로써 글쓴이의 의도를 간접적이나마 확인할 수 있음
- dplyr 패키지에 있는 count 함수 사용
  - ◆ tibble 형태의 word_space에 n이라는 빈도수 관련 변수 생성
    - ❖ n변수는 빈도수가 높은 순서대로 내림차순으로 정렬

```
word_space <- word_space %>%
 count(word, sort = T)
```

단어 빈도 분석하기

- 두 글자 이상으로 된 단어만 남기기
  - word_space의 word 변수에서 한 글자 단어는 의미 파악이 어려움
  - 따라서 두 글자 이상으로 구성된 단어만 남길 필요가 있음
    - str_count는 글자수를 세는 함수

```
word_space <- word_space %>%
 filter(str_count(word) > 1)
```

- 이상의 작업을 한 줄로 코딩하기

```
word_space <- word_space %>%
 count(word, sort = T) %>%
 filter(str_count(word) > 1)
```

단어 빈도 분석하기

- 빈도수 상위 20위 데이터 프레임 만들기
  - ◆ word_space는 빈도수 내림차순으로 정렬되어 있으므로, head 함수를 이용함

```
top20 <- word_space %>%
 head(20)
```

- 막대 그래프 그리기
  - ◆ geom_text: 막대 그래프에 빈도수 표시
  - ◆ labs & theme 그래프 제목 및 서식

```
ggplot(top20, aes(reorder(word, -n), n, fill = word)) + geom_bar(stat =
"identity") + geom_text(aes(label = n), hjust = -0.3) + labs(title =
"문재인 출마 연설문 단어 빈도") + theme(title = element_text(size = 12))
```

워드 클라우드 만들기

- 워드 클라우드란?
  - ◆ 단어 빈도를 구름 모양으로 표현한 그래프
  - ◆ 빈도에 따라 글자 크기와 색을 다르게 표현
  - ◆ 어떤 단어가 얼마나 많이 사용됐는지 한눈에 파악
- ggwordcloud 패키지에 있는 geom_text_wordcloud 함수 사용
  - ◆ geom_text_wordcloud는 난수(random number)를 사용하므로 seed 명령문 필요함

```
ggplot(word_space, aes(label = word, size = n)) +
 geom_text_wordcloud(seed = 1234) +
 scale_radius(limits = c(3, NA),      # 최소, 최대 단어 빈도
              range = c(3, 30))       # 최소, 최대 글자 크기
```

워드 클라우드 만들기

- 워드 클라우드 가다듬기
  - ◆ 참고할 사이트 - https://lepennec.github.io/ggwordcloud

```
ggplot(word_space,
       aes(label = word,
           size = n,
           col = n)) +                          # 빈도에 따라 색깔 표현
 geom_text_wordcloud(seed = 1234) +
 scale_radius(limits = c(3, NA),
              range = c(3, 30)) +
 scale_color_gradient(low = "#66aaf2",      # 최소 빈도 색깔
                      high = "#004EA1") +   # 최고 빈도 색깔
 theme_minimal()                            # 배경 없는 테마 적용
```

워드 클라우드 만들기

- 글자체 바꾸기
  - ◆ showtext 패키지 설치
  - ◆ 관련 사이트(https://fonts.google.com)에서 필요한 글자체 확인

```
install.packages("showtext")
library(showtext)

font_add_google(name = "Nanum Gothic", family = "nanumgothic")
showtext_auto()
```

  - ◆ Rstudio 실행할 때마다 글자체 설정해 주어야 함

ggplot 그래프 글자체 바꾸기

- Wordcloud 글자체 바꾸기

```
ggplot(word_space, aes(label = word, size = n, col = n)) +
geom_text_wordcloud(seed = 1234, family = "nanumgothic") + scale_radius
(limits = c(3, NA), range = c(3, 30)) + scale_color_gradient(low =
"#66aaf2", high = "#004EA1") + theme_minimal()
```

- ggplot 그래프 글자체 바꾸기

```
ggplot(top20, aes(reorder(word, -n), n, fill = word)) + geom_bar(stat =
"identity") + geom_text(aes(label = n), hjust = -0.3) + labs(title =
"문재인 출마 연설문 단어 빈도") + theme(title = element_text(size = 12),
text = element_text(family = "nanumgothic"))
```