텍스트 마이닝(2)

숙명여자대학교 경영학부 오중산

형태소 분석의 필요성

- 기존 ‘단어’ 중심 토큰화의 문제점
  - 띄어쓰기를 기준으로 하다 보니 의미 없는 단어가 포함됨
  - 예: ‘있습니다’, ‘합니다’ 등…
- 형태소 분석(Morphological Analysis)이란?
  - 형태소(morpheme)는 의미를 지니고 있는 가장 작은 말의 단위로, 더 나누면 의미가 없어짐
  - 문장에서 형태소를 추출해 명사, 동사, 형용사 등 품사로 분류하는 작업
  - 문장 내용 파악을 위해 ‘명사’가 중요함

형태소 분석 준비

- 형태소 분석을 위한 KoNLP 패키지 설치
  - 자바와 rJAVA 패키지 설치
    - install.packages("multilinguer")
    - library(multilinguer)
    - install_jdk()
  - KoNLP 의존성 패키지 설치
    - install.packages(c("stringr", "hash", "tau", "Sejong", "RSQLite", "devtools"), type = "binary")

형태소 분석 준비

- 형태소 분석을 위한 KoNLP 패키지 설치
  - KoNLP 패키지 설치
    - install.packages("remotes")
    - remotes::install_github("haven-jeon/KoNLP", upgrade = "never", INSTALL_opts=c("--no-multiarch"))
    - library(KoNLP)
  - 형태소 사전 설치하기
    - useNIADic()
    - KoNLP 설치 후에 한 번만 설치하면 됨

형태소 분석 준비

- KoNLP 패키지 설치가 안 될 경우
  - ◆ 방법1
    - ❖ 참고할 사이트 클릭!
  - ◆ 방법2
    - ❖ JAVA 설치(Windows 온라인): https://www.java.com/ko/download/manual.jsp
    - ❖ Rtools 4.0 설치: https://cran.r-project.org/bin/windows/Rtools/
  - ◆ 방법3
    - ❖ 다음 그림과 같은 메시지가 나오면 cli 패키지를 3.1.0으로 업데이트(설치)

```
** byte-compile and prepare package for lazy loading
loadNamespace(j <- i[[1L]], c(lib.loc, .libPaths()), versionCheck = vI[[j]])에서 다
음과 같은 에러가 발생했습니다:
  네임스페이스 'cli' 3.0.1는 이미 로드되었으나 >= 3.1.0가 필요합니다
```

명사를 기준으로 토큰화 실행

- unnest_tokens 함수 예시
  - ◆ library(tidytext)
  - ◆ text <- tibble(value = c("대한민국은 민주공화국이다.", "대한민국의 주권은 국민에게 있고, 모든 권력은 국민으로부터 나온다."))
  - ◆ text %>% unnest_tokens(input = value, output = word, token = extractNoun)

```
# A tibble: 7 x 1
  word
  <chr>
1 대한민국
2 민주공화국
3 대한민국
4 주권
5 국민
6 권력
7 국민
```

띄어쓰기 기준 추출

```
text %>%
  unnest_tokens(input = value,
                output = word,
                token = "words")
```

```
## # A tibble: 10 x 1
##    word
##    <chr>
##  1 대한민국은
##  2 민주공화국이다
##  3 대한민국의
##  4 주권은
##  5 국민에게
##  6 있고
##  7 모든
##  8 권력은
##  9 국민으로부터
## 10 나온다
```

명사를 기준으로 토큰화 실행

- unnest_tokens 실행
  - ◆ 앞서 만들어 놓은 moon 데이터 프레임에서 명사 기준으로 토큰화
    - ❖ word_noun <- moon %>%

      unnest_tokens(input = value, output = word,

      token = extractNoun)
  - ❖ 출력결과를 보면, 아직 KoNLP가 완전하지 않음을 알 수 있음

```
# A tibble: 1,757 x 1
   word
   <chr>
 1 "정권교체"
 2 "하겠습니"
 3 "정치"
 4 "교체"
 5 "하겠습니"
 6 "시대"
 7 "교체"
 8 "하겠습니"
 9 ""
10 "불비불명"
# ... with 1,747 more rows
```

명사 빈도 분석하기

- word_noun에서의 명사 빈도 분석하기
  - ◆ 빈도가 높을수록 해당 단어가 강조되었음
  - ◆ word_noun <- word_noun %>% count(word, sort = T) %>% filter(str_count(word) > 1)
    - ❖ 글자수가 2 이상인 명사에 대해 내림차순으로 빈도수 정리

```
# A tibble: 704 x 2
   word       n
   <chr>  <int>
 1 국민      21
 2 일자리    21
 3 나라      19
 4 우리      17
 5 경제      15
 6 사회      14
 7 성장      13
 8 대통령    12
 9 정치      12
10 하게      12
# ... with 694 more rows
```

막대 그래프 만들기

- 상위 20개 단어에 대한 막대 그래프 그리기
  - ◆ 글자체 선정
    - ❖ library(showtext)
    - ❖ font_add_google(name = "Black Han Sans", family = "BHS")
    - ❖ showtext_auto()
  - ◆ 막대 그래프 그리기
    - ❖ 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 명사 빈도") + theme(title = element_text(size = 12), text = element_text(family = "BHS"))

워드 클라우드 만들기

- word_noun에 대한 워드 클라우드 만들기
  - library(ggwordcloud)
  - ggplot(word_noun, aes(label = word, size = n, col = n)) + geom_text_wordcloud(seed = 1234, family = "BHS") + scale_radius(limits = c(2, NA), range = c(3, 15)) + scale_color_gradient(low = "darkgreen", high = "darkred") + theme_minimal()

특정 단어가 사용된 문장 살펴보기

- 문장 기준으로 토큰화
  - sentences_moon <- raw_moon %>% str_squish() %>% as_tibble() %>%

    unnest_tokens(input = value, output = sentence, token = "sentences")
  - 마침표를 기준으로 문장이 구분되므로, 특수문자 제거하지 않음

특정 단어가 사용된 문장 살펴보기

- 빈도수가 가장 많은 단어가 포함된 문장 확인하기
  - 빈도수가 가장 많은 단어(‘국민’과 ‘일자리’)를 포함한 문장을 str_detect 함수로 찾기
    - sentences_moon %>% filter(str_detect(sentence, "국민"))
    - sentences_moon %>% filter(str_detect(sentence, "일자리"))
    - 왼쪽 정렬로 모든 내용 출력하려면 %>% print.data.frame(right = F)를 붙임