텍스트 마이닝(5)

숙명여자대학교 경영학부 오중산

로그오즈비(log odds ratio) 로 단어 비교하기

- 로그오즈비의 정의와 이점
  - ◆ 로그오즈비는 오즈비에 자연로그를 취한 값
    - ❖ 상용로그는 밑이 10이지만, 자연로그는 밑이 무리수인 e(=2.71828182….)
  - ◆ 오즈비와 로그오즈비의 관계
    - ❖ 오즈비 $= 1 \rightarrow$ 로그오즈비 $= 0$
    - ❖ 오즈비 $> 1 \rightarrow$ 로그오즈비 > 0
    - ❖ 오즈비 $< 1 \rightarrow$ 로그오즈비 < 0
  - ◆ 로그오즈비 막대그래프는 텍스트에서의 단어 비교하기를 더욱 명확하게 함

로그오즈비(log odds ratio) 로 단어 비교하기

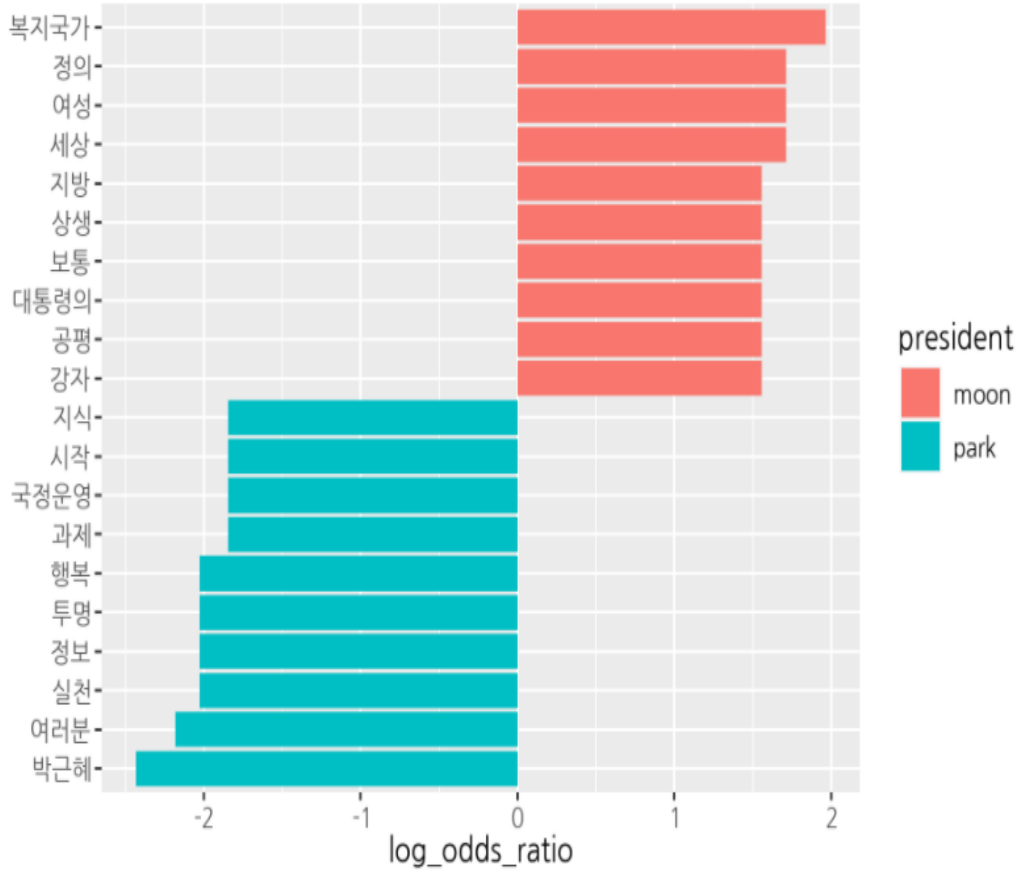
- 로그오즈비 구하기
  - ◆ frequency_wide에서 로그오즈비 구하기
    - ❖ frequency_wide <- frequency_wide %>% mutate(log_odds_ratio = log(odds_ratio))
  - ◆ 로그오즈비값과 텍스트에서의 단어의 비중
    - ❖ 로그오즈비가 클수록 moon에서 비중이 큰 단어이고, 작을수록 park에서 비중이 큰 단어
    - ❖ 로그오즈비가 0이면 양쪽에서 비중이 비슷한 단어

로그오즈비(log odds ratio) 로 단어 비교하기

- 로그오즈비를 이용해서 중요한 단어 비교하기
  - ◆ 두 연설문(moon과 park)에서 로그오즈비 상위 10개 단어 추출하기
    - ❖ top10 <- frequency_wide %>% group_by(president = ifelse(log_odds_ratio > 0, "moon", "park")) %>% slice_max(abs(log_odds_ratio), n = 10, with_ties = F)
    - ❖ mutate가 아닌 group_by를 쓴 것에 주의해야 함!
      - ✓ president라는 새로운 변수를 만들고,
      - ✓ president에 따라 집단을 두 개(moon / park)로 구분한 후,
      - ✓ 각 집단별로 로그오즈비 절대값이 큰 상위 10개 사례 추출
    - ❖ rank 함수를 이용해서 오즈비 기준 상·하위 10개 사례 추출 결과와 동일함

로그오즈비(log odds ratio) 로 단어 비교하기

- 막대 그래프 만들기
  - ◆ 두 연설문에 대해 하나의 그래프로 만듦
    - ❖ ggplot(top10, aes(reorder(word, log_odds_ratio), log_odds_ratio, fill = president)) + geom_bar(stat = "identity") + coord_flip() + labs(x = NULL)
    - ❖ reorder_within 함수나 facet_wrap 함수 필요 없음!



TF-IDF: 여러 텍스트의 단어 비교하기

- 오즈비나 로그오즈비의 한계
  - 서로 다른 두 텍스트에서 단어의 상대적 비중 차이를 알 수 있지만, 세 개 이상 텍스트를 대상으로 비교할 수 없음
- TF-IDF
  - 중요한 단어는 1)흔하지 않으면서 2)특정 텍스트에서 많이 사용되는 특징이 있음
  - TF-IDF는 여러 텍스트를 비교하여 특정 텍스트에서 어떤 단어가 자주 사용되었는지 알려주는 지표
    - 텍스트별로 단어별 TF-IDF 값 계산

TF-IDF: 여러 텍스트의 단어 비교하기

- TF-IDF
  - ◆ TF(term frequency)란?
    - ❖ 특정 단어가 특정 텍스트에서 사용된 회수(단어 빈도)
  - ◆ DF(document frequency)란?
    - ❖ 특정 단어가 사용된 텍스트 수(문서 빈도)
    - ❖ IDF(Inverse document frequency)는 DF의 역수에 문서 개수(N)을 곱한 후 자연로그를 취한값
      - ✓ IDF = log(N / DF)
    - ❖ IDF가 크다는 것은 특정 단어가 특정 텍스트(들)에서만 사용된다는 것을 의미하며, IDF가 작을 경우 여러 텍스트(들)에서 흔하게 사용된다는 것을 의미

TF-IDF: 여러 텍스트의 단어 비교하기

- TF-IDF
  - ◆ TF-IDF = TF × IDF = TF× log(N / DF)
  - ◆ 흔하지 않으면서 특정 텍스트에서만 많이 나오는 단어일수록 TF-IDF값이 높음
  - ◆ TF가 커도 IDF가 0이면 TF-IDF도 0이 되는 문제는 weighted log odds가 대안

TF

단어	자기소개서 A	자기소개서 B	자기소개서 C
저는	15	10	10
스카이다이빙	3	0	0
자기주도적	3	5	3
데이터	0	5	1
배낭여행	2	3	5

IDF

단어	DF	IDF
저는	3	$\log\frac{3}{3}=0$
스카이다이빙	1	$\log\frac{3}{1}=1.1$
자기주도적	3	$\log\frac{3}{3}=0$
데이터	2	$\log\frac{3}{2}=0.4$
배낭여행	3	$\log\frac{3}{3}=0$

TF-IDF

단어	자기소개서 A	자기소개서 B	자기소개서 C
저는	$15\times\log\frac{3}{3}=0$	$10\times\log\frac{3}{3}=0$	$10\times\log\frac{3}{3}=0$
스카이다이빙	$3\times\log\frac{3}{1}=3.3$	$0\times\log\frac{3}{1}=0$	$0\times\log\frac{3}{1}=0$
자기주도적	$3\times\log\frac{3}{3}=0$	$5\times\log\frac{3}{3}=0$	$3\times\log\frac{3}{3}=0$
데이터	$0\times\log\frac{3}{2}=0$	$5\times\log\frac{3}{2}=2$	$1\times\log\frac{3}{2}=0.4$
배낭여행	$2\times\log\frac{3}{3}=0$	$3\times\log\frac{3}{3}=0$	$5\times\log\frac{3}{3}=0$

TF-IDF: 여러 텍스트의 단어 비교하기

- TF-IDF 구하기
  - ◆ 필요한 텍스트 데이터
    - ❖ 역대 대통령(4명)의 출마 선언문 파일(speeches_presidents.csv)
  - ◆ readr 패키지에 있는 read_csv 함수 이용
    - ❖ csv 파일을 불러와서 tibble로 바꾸어 주는 동시에, 데이터 불러오는 속도도 빠름
    - ❖ raw_speeches <- read_csv("speeches_presidents.csv")
  - ◆ 전처리, 토큰화(명사 기준), 단어 빈도 구하기
    - ❖ speeches <- raw_speeches %>% mutate(value = str_replace_all(value, "[^가-힣]", " "), value = str_squish(value))
    - ❖ speeches <- speeches %>% unnest_tokens(input = value, output = word, token = extractNoun)
    - ❖ frequency_four <- speeches %>% count(president, word) %>% filter(str_count(word) > 1)

TF-IDF: 여러 텍스트의 단어 비교하기

- TF-IDF 구하기
  - ◆ 연설문별로 TF, IDF, and TF-IDF 계산하기
    - ❖ tidytext 패키지에 있는 bind_tf_idf(term = , document = , n =) 함수 사용
    - ❖ term은 TF-IDF 대상 단어 변수, document는 텍스트(연설문) 구분 변수, n은 단어 빈도 변수
    - ❖ frequency_four <- frequency_four %>% bind_tf_idf(term = word, document = president, n = n) %>% arrange(-tf_idf)
    - ❖ 주의!: TF값은 특정 텍스트에서 특정 단어의 비중(특정 텍스트에서 특정 단어 빈도수 / 특정 텍스트의 전체 단어 빈도수)
    - ❖ 대통령 연설문별로 TF-IDF가 높은 단어 확인 가능
      - ✓ frequency_four %>% filter(president == "문재인 / 박근혜 / 이명박 / 노무현")

TF-IDF: 여러 텍스트의 단어 비교하기

- TF-IDF 막대 그래프 그리기
  - ◆ 대통령별로 TF-IDF가 높은 10개 단어 추출하기
    - ❖ top10 <- frequency_four %>% group_by(president) %>% slice_max(tf_idf, n = 10, with_ties = F)
  - ◆ 그래프 순서 정하기
    - ❖ top10$president <- factor(top10$president, levels = c("문재인", "박근혜", "이명박", "노무현"))
  - ◆ 막대 그래프 그리기
    - ❖ ggplot(top10, aes(reorder_within(word, tf_idf, president), tf_idf, fill = president)) + geom_bar(stat = "identity") + coord_flip() + facet_wrap(~ president, scales = "free") + scale_x_reordered() + labs(x = NULL)