텍스트 마이닝(6)

숙명여자대학교 경영학부 오중산

감정 사전 활용하기

- 감정분석(sentiment analysis)이란?
  - 텍스트에 어떤 감정이 담겨 있는지 분석하는 방법
    - 긍정적인 감정인가, 아니면 부정적인 감정인가?
- 감정 사전이란?
  - 감정 단어와 감정의 강도를 표현한 숫자로 구성된 사전
    - 사전을 이용해서 문장의 단어에 감정 점수를 부여한 후에 합산

감정 사전 활용하기

- KNU 한국어 감성 사전
  - ◆ 국립 군산대학교 소프트웨어융합공학과에서 개발한 감정 사전
  - ◆ dic <- read_csv("knu_sentiment_lexicon.csv")
    - ❖ word: 감정 단어
      - ✓ 단일어(한 단어로 구성), 복합어(두 개 이상 단어로 구성), 이모티콘
      - ✓ 총 14,854개 단어로 구성됨
    - ❖ polarity: 감정의 강도
      - ✓ +2~-2의 5개 정수로 구성되며, 긍정단어는 +로 부정단어는 –로 중성단어는 0으로 표시

감정 사전 활용하기

- KNU 한국어 감성 사전 살펴보기
  - ◆ 이모티콘 살펴보기
    - ❖ library(stringr) / dic %>% filter(!str_detect(word, "[가-힣]")) %>% arrange(word)
  - ◆ 14,854개 단어 분류
    - ❖ dic %>% mutate(sentiment = ifelse(polarity >= 1, "pos", ifelse(polarity <= -1, "neg", "neu"))) %>% count(sentiment)

```
# A tibble: 3 x 2
  sentiment     n
  <chr>     <int>
1 neg        9829
2 neu         154
3 pos        4871
```

감정 사전 활용하기

- 문장의 감정 점수 구하기

  ◆ STEP1: 단어 기준으로 토큰화하기

  ❖ df <- tibble(sentence = c("디자인 예쁘고 마감도 좋아서 만족스럽다.", "디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다."))

  ❖ library(tidytext) / df <- df %>% unnest_tokens(input = sentence, output = word, token = "words", drop = F)

  ✓ drop = F는 원문(sentence 변수)을 제거하지 않는다는 의미

```
# A tibble: 12 x 2
   sentence                                          word
   <chr>                                             <chr>
 1 디자인 예쁘고 마감도 좋아서 만족스럽다.           디자인
 2 디자인 예쁘고 마감도 좋아서 만족스럽다.           예쁘고
 3 디자인 예쁘고 마감도 좋아서 만족스럽다.           마감도
 4 디자인 예쁘고 마감도 좋아서 만족스럽다.           좋아서
 5 디자인 예쁘고 마감도 좋아서 만족스럽다.           만족스럽다
 6 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 디자인은
 7 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 괜찮다
 8 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 그런데
 9 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 마감이
10 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 나쁘고
11 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 가격도
12 디자인은 괜찮다. 그런데 마감이 나쁘고 가격도 비싸다. 비싸다
```

감정 사전 활용하기

- 문장의 감정 점수 구하기
  - ◆ STEP2: 단어에 감정 점수 부여하기
    - ❖ word 기준으로 left_join을 이용하여 감정 사전 결합
    - ❖ 만약 단어가 감정 사전에 없으면 NA가 되는데, 이때는 0으로 변경
    - ❖ df <- left_join(df, dic, by = "word") %>% mutate(polarity = ifelse(is.na(polarity), 0, polarity))
  - ◆ STEP3: 문장별로 감정 점수 합산하기
    - ❖ 합산 점수가 양수면, 문장에서 긍정적인 단어가 상대적으로 많이 사용되었음을 의미함
    - ❖ score_df <- df %>% group_by(sentence) %>% summarise(score = sum(polarity))

댓글 감정 분석하기

- 기본적인 전처리
  - ◆ 기사 댓글 소개
    - ❖ “news_comment_parasite.csv”에는 영화 기생충의 아카데미상 수상 관련 댓글을 담고 있음
    - ❖ raw_news_comment <- read_csv("news_comment_parasite.csv")
  - ◆ 고유 번호 변수(id) 만들기와 html 특수 문자 제거하기
    - ❖ install.packages("textclean") / library(textclean)
      - ✓ 웹에서 만든 텍스트의 html 특수문자 제거를 위한 replace_html 함수는 textclean 패키지에 있음
    - ❖ news_comment <- raw_news_comment %>% mutate(id = row_number(), reply = str_squish(replace_html(reply)))

댓글 감정 분석하기

- 단어 기준 토큰화 및 감정 점수 부여하기
  - 토큰화
    - word_comment <- news_comment %>% unnest_tokens(input = reply, output = word, token = "words", drop = F)
  - 감정 점수 부여하기
    - word_comment <- left_join(word_comment, dic, by = "word") %>% mutate(polarity = ifelse(is.na(polarity), 0, polarity))

댓글 감정 분석하기

- 감정 분류 및 단어 빈도별 막대 그래프 그리기
  - ◆ 감정 분류
    - ❖ word_comment <- word_comment %>% mutate(sentiment = ifelse(polarity == 2, "pos", ifelse(polarity == -2, "neg", "neu")))
      - ✓ 문장에 대해 감정 분류 기준값을 ±2로 강화함
  - ◆ 빈도수 상위 10개 단어 데이터 프레임 만들기
    - ❖ top10_sentiment <- word_comment %>% filter(sentiment != "neu") %>% count(sentiment, word) %>% group_by(sentiment) %>% slice_max(n, n = 10)
      - ✓ count 함수에 의해 n(빈도수) 변수를 만들고, 감정에 따른 문장 유형에 따라 상위 10개 단어 추출
      - ✓ 빈도수 동률인 단어를 모두 포함

댓글 감정 분석하기

- 감정 분류 및 단어 빈도별 막대 그래프 그리기
  - ◆ 빈도수 상위 10개 단어에 대한 막대 그래프 그리기
    - ❖ ggplot(top10_sentiment, aes(reorder_within(word, n, sentiment), n, fill = sentiment)) + geom_bar(stat = "identity") + coord_flip() + facet_wrap(~sentiment, scales = "free") + scale_x_reordered() + labs(x = NULL) + geom_text(aes(label = n))
    - ❖ geom_text 함수를 통해 그래프에 빈도수를 수치로 표기

댓글 감정 분석하기

- 댓글별 감정 점수 구하고 내용 살펴보기
  - ◆ 댓글별 감정 점수 구하기
    - ❖ score_comment <- word_comment %>% group_by(id, reply) %>% summarise(score = sum(polarity)) %>% ungroup()
      - ✓ word_comment에서 댓글(reply)이 동일하더라도 id가 다르면 서로 다른 댓글로 취급하기 위해 id와 reply로 집단을 구분함
      - ✓ ungroup 함수를 이용하여 그룹을 해제함
  - ◆ 감정 점수 상/하위 10개 댓글 확인하기
    - ❖ score_comment %>% arrange(-score) %>% head(10)
    - ❖ score_comment %>% arrange(score) %>% head(10)

댓글 감정 분석하기

- 댓글별 감정 점수 구하고 내용 살펴보기
  - 감정 점수별 빈도 구하기
    - score_comment %>% count(score)
  - 감정 점수에 따른 감정 분류 및 유형별 빈도수 확인하기
    - score_comment <- score_comment %>% mutate(sentiment = ifelse(score >= 1, "pos", ifelse(score <= -1, "neg", "neu")))
    - score_comment %>% count(sentiment)