ch1 워드 클라우드

[4 조]

2022 5 13

최근 육류를 대체하는 대체 식품이 늘어나면서 비건의 관심도 또한 증가하였습니다.

이에 따른 육류 소비량과 환경의 연관 관계 또한 집중되고 있어 R 아조 팀에서는 비건이 환경에 미치는 영향과 비거니즘의 특성에 대해 알아보고자 합니다.

먼저 '비건' 키워드를 가지고 어떤 단어가 많이 나오는지 워드 클라우드를 통해 분석해 보았습니다.

워드 클라우드는 트위터에서 비건을 검색했을 때 많이 나오는 단어들을 빈도수 별로 크기를 다르게 하여 표현하였습니다. 빈도수가 많을수록 단어의 크기가 커집니다

라이브러리 로드

```
library(rtweet)
library(twitteR)
library(stringr)
library(KoNLP)
library(wordcloud)
library(dplyr)
library(stringr)
library(devtools)
library(wordcloud2)
```

트위터 API 사용

```
api_key <- "dwipwvG38SWnupb7nIcv3WiO7"
api_secret <- "5AaeGt6k8bjHhzP69GBvxFV6bP5g9xDzmO3c8NzmGjLZGZ8voe"
access_token <- "1426214458481733638-3kD1CA0vPkHMCvfsmsl5uPNAIsB8Jn"
access_secret <- "vOj0GEKHjBS8CbW9yU7MEqN6TsgPWRyuWXXiz5Dd4F2rM"

setup_twitter_oauth(api_key,api_secret,access_token,access_secret)
```

```
## [1] "Using direct authentication"
```

트위터에 '비건' 검색 결과 추출

keyword <- enc2utf8("비건")
vegan <- searchTwitter(keyword,n=1500,lang="ko",resultType="recent")
head(vegan)

위와 같이 진행 시, 트위터 검색 결과가 실시간으로 달라지고 시간이 너무 오래걸리는 이슈 발생.

따라서, 아래와 같이 데이터를 저장하고 테스트 시에 그대로 저장한 데이터를 불러오는 방법 이용.

```
#save(vegan, "vegan_raw.RData")
load("vegan_raw.RData")
```

트위터 정보에서 text 부분만 추출

```
vegan_df <- twListToDF(vegan)
head(vegan_df, 3)

##
                                                                    text
## 1 RT @donghaemul_kr: 오늘 저녁엔 참치 없는 '비건 참치' 먹고, 탈육식 함께해요!\n\n#세계참치의날 #참치 #참치마요 #비건참치 #식물성참치 #대체육 #탈육식 #비건세상만들기 #동물해방물결 #느끼는모두에게자유를 https://t.co/r…
## 2                                  RT @MX_rang: 나 이 mbb 알아 비건하고 불독티 좋아하고 해장을 런닝으로 하는 몽베베자나 https://t.co/zqYr4kVOVV
## 3                                          &lt;메종&gt; 에디터들의 내돈내산 비건 식당 리뷰 (출처 : 메종 | 네이버 포스트) https://t.co/fF0MVj0r1g
##   favorited favoriteCount replyToSN             created truncated replyToSID
## 1     FALSE             0      <NA> 2022-05-02 10:24:49     FALSE       <NA>
## 2     FALSE             0      <NA> 2022-05-02 10:24:33     FALSE       <NA>
## 3     FALSE             0      <NA> 2022-05-02 10:23:55     FALSE       <N
```

```
A>
##                    id replyToUID
## 1 1521073215308648448       <NA>
## 2 1521073148854435840       <NA>
## 3 1521072989885722624       <NA>
##
statusSource
## 1 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for
 Android</a>
## 2   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter fo
r iPhone</a>
## 3 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for
 Android</a>
##      screenName retweetCount isRetweet retweeted longitude latitude
## 1     cute_aram            9      TRUE     FALSE        NA       NA
## 2 poorhungryant           75      TRUE     FALSE        NA       NA
## 3     mango4988            0     FALSE     FALSE        NA       NA

vegan_txt_list <- vegan_df$text
```

Emoji 제거

```
#write(vegan_txt_list, "Emoji_rm.txt")
vegan_txt_list <- readLines("Emoji_rm.txt")
```

Emoji 가 <U+0000> 형식이라 R 내부에서 warning message 발생.

R 에는 emoji 를 지우는 함수가 존재하지 않아서, txt 파일로 내보낸 후 정제.

사전 불러오기 및 사전에 단어 추가

```
useNIADic()

## Backup was just finished!
## 1213109 words dictionary was built.

mergeUserDic(data.frame(c("비건","비거니즘","논비건","미라클버거","요거트","풀무
원","종달리","낫아워스","불매","레시피", "일기","롯데리아", "인증마크","참나물잣페
스토","페스토"),"ncn"))

## 15 words were added to dic_user.txt.
```

text 에서 명사만 추출

```
vegan_noun <- sapply(vegan_txt_list, extractNoun,USE.NAMES=F)

vegan_noun_1 <- unlist(vegan_noun)
str(vegan_noun_1)

##  chr [1:20638] "RT" "donghaemul" "kr" "오늘" "저녁" "참치" "참치" "육식"
"" ...
```

필요없는 부분 제거

```
#vegan_noun_1 <- str_replace_all(vegan_noun_1, "\\W", " ")
vegan_noun_1 <- gsub('[[:upper:]]',"",vegan_noun_1) #영어 대문자 제거

vegan_noun_1 <- gsub('[[:lower:]]',"",vegan_noun_1) #영어 소문자 제거
vegan_noun_1 <- gsub('[[:punct:]]',"",vegan_noun_1) #특수문자 제거
vegan_noun_1 <- gsub('\\d',"",vegan_noun_1) #숫자 제거
vegan_noun_1 <- gsub('\\n',"",vegan_noun_1) #엔터 제거
vegan_noun_1 <- gsub('[ㄱ-ㅎ]',"",vegan_noun_1) #자음으로만 된 글자 제거
vegan_noun_1 <- gsub('[ㅏ-ㅣ]',"",vegan_noun_1) #모음으로만 된 글자 제거
head(vegan_noun_1)

## [1] ""     ""     ""     "오늘" "저녁" "참치"
```

관련 없는 단어 전처리

```
txt <- readLines("vegan_gsub.txt", encoding="UTF-8")
cnt_txt <- length(txt)
for(i in 1:cnt_txt){
  vegan_noun_1 <- gsub((txt[i]), "", vegan_noun_1)
}
str(vegan_noun_1)

##  chr [1:20638] "" "" "" "" "저녁" "참치" "참치" "육식" "" "세계참치의날" ...

###공백 제거
write(unlist(vegan_noun_1),"after_gsub.txt")
vegan_noun_2 <- read.table("after_gsub.txt")
str(vegan_noun_2)
```

```
## 'data.frame':    13190 obs. of  1 variable:
##  $ V1: chr  "저녁" "참치" "참치" "육식" ...
```

###두 단어 이상만 추출

```
noun_two <- filter(vegan_noun_2, nchar(V1)>=2)
head(noun_two)
```

```
##             V1
## 1         저녁
## 2         참치
## 3         참치
## 4         육식
## 5 세계참치의날
## 6         참치
```

단어별 빈도 수 카운트

```
wordcount <- table(noun_two)
head(noun_two)
```

```
##             V1
## 1         저녁
## 2         참치
## 3         참치
## 4         육식
## 5 세계참치의날
## 6         참치
```

데이터 프레임으로 변환

```
df_word <- as.data.frame(wordcount)
str(df_word)
```

```
## 'data.frame':    2152 obs. of  2 variables:
##  $ noun_two: Factor w/ 2152 levels "가게","가격",..: 1 2 3 4 5 6 7 8 9 10
 ...
##  $ Freq    : int  3 3 5 1 1 1 4 3 3 1 ...
```

빈도 수 상위 50 개 추출

```
top_50 <- df_word %>%
  arrange(desc(Freq)) %>%
  head(50)
head(top_50)
```

```
##           noun_two Freq
## 1             비건  432
## 2           논비건  200
## 3 나의비거니즘일기  162
## 4           요거트  152
## 5             소스  139
## 6           식물성  126
```

Word Cloud 시각화

```
palete <- brewer.pal(9, "Set1")
wordcloud(words=top_50$noun_two,
          freq=top_50$Freq,
          min.freq=2,
          max.words=50,
          random.order = F,
          rot.per=0.07,
          scale=c(8,0.3),
          colors=palete)
```


비건
나의비거니즘일기
진
소스
요거트
식물성
롯데
언니네텃밭
농업
제품
음식
미라클버거
페스토
신제품
토마토
여성
생태
먹거리
곡물
롯데마트
풀무원
베이컨
레지피
참나물잣페스토
브랜드
화장품
크림
도시락
채소
카페
소비
바질
요리
디엠
지구
지향
크림치즈
간편
월요일
식당냉동식품
화장
가지절임
식사

Word Cloud2 시각화

```
wordcloud2(data=top_50,color = "random-light")
```