

제10장 네이버 Open API 활용

제10장 네이버 Open API 활용

1. Naver Open API를 통한 블로그 검색 => 데이터 확보
 - 1-1. 블로그 검색
 - 1-2. 블로그 내용 리스트 만들기
2. 확보된 데이터의 시각화 : 1) 차트 작성, 2) 워드 클라우드
 - 2-1. 데이터 전처리 : 1) 각 행의 단어 추출
 - 2-2. 명사별 도수분포표 작성 :
 - 1) 불필요한 단어 삭제
 - 2) 필요한 단어 추가
 - 3) 명사별 도수분포표 작성 다시 확인
 - 2-3. wordcount의 차트 작성
 - 2-4. wordcloud 작성

1. Naver Open API를 통한 블로그 검색 => 데이터 확보

```
install.packages("RCurl")
install.packages("XML")
library(RCurl)
library(XML)

#-----
# 네이버 API url
# 1) blog url : https://openapi.naver.com/v1/search/blog.xml
# 2) news url : https://openapi.naver.com/v1/search/news.xml
#-----
searchUrl <- "https://openapi.naver.com/v1/search/blog.xml"

#-----
# 네이버 API 인증키 설정
# https://developers.naver.com/apps/#/list 에서 확인
#-----
Client_ID <- "yv...gv"
Client_Secret <- "qb...PJ"
```

1-1. 블로그 검색

```
no_display <- 20; # 검색결과의 페이지 수 : 20
query <- URLEncode(iconv("여름추천요리", "euc-kr", "UTF-8")) # iconv() : 키워드 '여름추천요리'의 한글코드 변환
url <- paste(searchUrl, "?query=", query, "&display=", no_display, sep="")

doc <- getURL(url,
  httpheader = c('Content-Type' = "application/xml",
    'X-Naver-Client-Id' = Client_ID,
    'X-Naver-Client-Secret' = Client_Secret))
```

1-2. 블로그 내용 리스트 만들기

=> 내용 한 줄을 하나의 문자 벡터 요소로 만들기

```
doc2 <- htmlParse(doc, encoding="UTF-8")
text <- xpathSApply(doc2, "//item/description", xmlValue)
text
```

2. 확보된 데이터의 시각화: 1) 차트 작성, 2) 워드 클라우드

(8장 워드 클라우드 참조)

2-1. 데이터 전처리: 1) 각 행의 단어 추출

```
# install.packages("KoNLP")
# install.packages("RColorBrewer")
# install.packages("wordcloud")
library(KoNLP)
library(RColorBrewer)
library(wordcloud)

useSejongDic()

noun <- sapply(text, extractNoun, USE.NAMES=F) # 각 행의 단어 추출
noun

# p.275
noun2 <- unlist(noun) # 추출된 단어들을 벡터로 통합
noun2
```

2-2. 명사별 도수분포표 작성:

다음의 (2-1)~(2-3)을 반복 수행한다.

```
wordcount <- table(noun2) # 단어 빈도수 (도수 분포표) 파악
wordcount
```

1) 불필요한 단어 삭제

```
noun2 <- Filter(function(x){nchar(x) >= 2}, noun2)
noun2

# p.277
noun2 <- gsub('\\d+', '', noun2)
noun2 <- gsub('<b>', '', noun2)
```

```

noun2 <- gsub('</b>', '', noun2)
noun2 <- gsub('&amp;', '', noun2)
noun2 <- gsub('&lt;', '', noun2)
noun2 <- gsub('&gt;', '', noun2)
noun2 <- gsub('&quot;', '', noun2)
noun2 <- gsub('\"', '', noun2)
noun2 <- gsub('\\', '', noun2)
noun2 <- gsub(' ', '', noun2)
noun2 <- gsub('-', '', noun2)
# noun2 <- gsub('++', '', noun2) # 불필요한 '++' 제거

noun2

```

2) 필요한 단어 추가

```

mergeUserDic(data.frame(c(""), c("ncn"))) # 추가할 단어 기입

```

3) 명사별 도수분포표 작성 다시 확인

```

wordcount <- table(noun2)
head(sort(wordcount, decreasing=T), 30)

```

2-3. wordcount의 차트 작성

```

temp <- sort(wordcount, decreasing=T)[1:30] # 내림차순(빈도가 가장 많은 것에서 부터 가장 작
은 순)으로 단어 정렬, 상위 30개 선택
temp                                     # 확인

temp <- temp[-1]                        # 공백단어 제거
barplot(temp, las = 2, names.arg = names(temp), # 차트 출력
        col = rainbow(30), main = "Most frequent words", # 축, 제목 입력
        ylab = "Word frequencies", ylim = c(0,10)) # 축 입력

```

2-4. wordcloud 작성

```

palette <- brewer.pal(9, "Set1")

wordcloud(names(wordcount),
          freq=wordcount,
          scale=c(5,0.25),
          rot.per=0.25,
          min.freq=1,
          random.order=F,
          random.color=T,
          colors=palette)

```



