MOVIE_CRAWLING

GI, SEUNG

2019 년 3 월 26 일

#영화 댓글 크롤링

###**Crawling** 크롤링(crawling) 혹은 스크레이핑(scraping)은 웹 페이지를 그대로 가져와서 거기서 데이터를 추출해 내는 행위이다.

필요한 패키지 설치

```
install.packages('rvest')
install.packages('dplyr')
install.packages('stringr')
```

##1.영화 페이지 크롤링

```
movie_url='https://movie.naver.com/movie/running/current.nhn?view=list&tab=normal&order=reserve'
```

네이버영화

##2. 상위 10 개 영화 크롤링

- **영화 링크 크롤링**

```
require(rvest)
```

```
## Loading required package: rvest
```

```
## Warning: package 'rvest' was built under R version 3.5.3
```

```
## Loading required package: xml2
```

```
## Warning: package 'xml2' was built under R version 3.5.2
```

```
html=read_html(movie_url)
html_1=html_nodes(html,'.tit')
html_2=html_nodes(html_1,'a')
html_3=html_attr(html_2,'href')
head(html_3)
```

```
## [1] "/movie/bi/mi/basic.nhn?code=161967"
## [2] "/movie/bi/mi/basic.nhn?code=163788"
```

```
## [3] "/movie/bi/mi/basic.nhn?code=97631"
## [4] "/movie/bi/mi/basic.nhn?code=164125"
## [5] "/movie/bi/mi/basic.nhn?code=18781"
## [6] "/movie/bi/mi/basic.nhn?code=136900"
```

- 위의 코드를 한줄로 합쳐준다.

```
require(dplyr)

## Loading required package: dplyr

## Warning: package 'dplyr' was built under R version 3.5.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

html=read_html(movie_url)
movie_link=html %>% html_nodes('.tit') %>% html_nodes('a') %>% html_attr('hre
f')
head(movie_link)

## [1] "/movie/bi/mi/basic.nhn?code=161967"
## [2] "/movie/bi/mi/basic.nhn?code=163788"
## [3] "/movie/bi/mi/basic.nhn?code=97631"
## [4] "/movie/bi/mi/basic.nhn?code=164125"
## [5] "/movie/bi/mi/basic.nhn?code=18781"
## [6] "/movie/bi/mi/basic.nhn?code=136900"
```

- 영화 코드 추출

```
require(stringr)

## Loading required package: stringr

## Warning: package 'stringr' was built under R version 3.5.3

n=10  #분석할 영화의 개수
movie_code=str_sub(movie_link[1:n],29,-1)
movie_code

##  [1] "161967" "163788" "97631"  "164125" "18781"  "136900" "181704"
##  [8] "183132" "177967" "176040"
```

- **영화 이름 추출**

```
m_name=html %>% html_nodes('.tit') %>% html_nodes('a')
class(m_name)
```

```
## [1] "xml_nodeset"
```

```
#인덱싱 해주기 위해 character 로 바꿔준다.
movie_name=vector()
for (i in 1:n){
  name=as.character(m_name[i])
  movie_name[i]=str_sub(name,46,-5)
}
movie_name
```

```
##  [1] "기생충"                "알라딘"                " 인 블랙: 인터내셔널"
##  [4] "엑스맨: 다크 피닉스"  "웃집 토토로"          "어벤져스: 엔드게임"
##  [7] "로켓맨"                "교회오빠"              "악인전"
## [10] "파리의 딜릴리"
```

##3. 각 영화의 url 만들기

- **영화의 댓글을 보기 위해서는 각 영화의 페이지로 들어가야한다.**

```
Each_movie=vector()
for (i in 1:n){
  Each_movie[i]=paste('https://movie.naver.com/movie/bi/mi/point.nhn?code=',
                      movie_code[i],'#tab',sep='')
}
head(Each_movie)
```

```
## [1] "https://movie.naver.com/movie/bi/mi/point.nhn?code=161967#tab"
## [2] "https://movie.naver.com/movie/bi/mi/point.nhn?code=163788#tab"
## [3] "https://movie.naver.com/movie/bi/mi/point.nhn?code=97631#tab"
## [4] "https://movie.naver.com/movie/bi/mi/point.nhn?code=164125#tab"
## [5] "https://movie.naver.com/movie/bi/mi/point.nhn?code=18781#tab"
## [6] "https://movie.naver.com/movie/bi/mi/point.nhn?code=136900#tab"
```

네이버영화 평점 1 위

##4. 댓글 URL 만들기

- **각 영화 페이지에서 댓글프레임의 URL 구조를 만든다.**

```
reple_html=read_html(Each_movie[1])
reple_frame=reple_html %>% html_nodes('.ifr_module2') %>% html_nodes('iframe')
 %>% html_attr('src')
head(reple_frame)
```

```
## [1] "/movie/bi/mi/pointWriteFormList.nhn?code=161967&type=after&isActualPo
intWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptio
nReject=false"
```

######https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=163608&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&page=1

URL 의 구조가 규칙적이다. 각 영화에서 댓글창의 다음페이지로 넘어가려면 맨끝에 페이지숫자를 바꿔주면된다. 다른영화의 댓글로 넘어가려면 첫줄의 코드값을 위에서 만든 movie_code 값으로 바꿔준다.

- **댓글 페이지 url 크롤링**

```
m=50 #분석할 댓글의 페이지 수
reple_url=list()
page=vector()
reple_frame

## [1] "/movie/bi/mi/pointWriteFormList.nhn?code=161967&type=after&isActualPo
intWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptio
nReject=false"

for (i in 1:n){
  frame=paste0(str_sub(reple_frame,1,41),movie_code[i],str_sub(reple_frame,48,
-1))
  for (j in 1:m){
    page[j]=paste('https://movie.naver.com',
                  reple_frame,'&page=',j,sep='')
  }
  reple_url[[i]]=page
}
names(reple_url)=movie_name
head(reple_url[[3]])

## [1] "https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=16196
7&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=fal
se&isMileageSubscriptionReject=false&page=1"
## [2] "https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=16196
7&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=fal
se&isMileageSubscriptionReject=false&page=2"
## [3] "https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=16196
7&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=fal
se&isMileageSubscriptionReject=false&page=3"
## [4] "https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=16196
7&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=fal
se&isMileageSubscriptionReject=false&page=4"
## [5] "https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=16196
```

```
7&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=fal
se&isMileageSubscriptionReject=false&page=5"
## [6] "https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=16196
7&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=fal
se&isMileageSubscriptionReject=false&page=6"
```

네이버영화 평점 1 위 댓글

##5.댓글 크롤링

- **EXAMPLE**

```
ex_reple_html=read_html(reple_url[[1]][1])
ex_reple=ex_reple_html %>% html_nodes('.score_reple') %>% html_nodes('p')
head(ex_reple)
```

```
## {xml_nodeset (6)}
## [1] <p>비에 젖지 않는 고급 장난감 텐트와, 비에 젖다 못해 잠기고 마는 반지하 가
구  </p>
## [2] <p>최근 본 영화중 가장 충격적이었음... 근데 보니까 15 세말고 19 세 걸어야될
것같던데..  </p>
## [3] <p>지하철이라는 단어가 언급되는 순간, 대다수의 관객은 자신이 어디에 이입할
지를 안다.  </p>
## [4] <p>전 가정부가 집 벨 누를 때 부터 이 영화는 장르가 바뀐다... 역대급 꿀잼영
화  </p>
## [5] <p>황금종려상 수상작을 자막 없이 볼 수 있다는 것 자체로 좋다.  </p>
## [6] <p>뭔가 모를 불쾌한 영화였다. 영화가 불쾌하다는게 아니라 보는 내내 가슴에
뭔가 영화에서 나오는 수석이 얹혀져있는 영화 ...
```

태그'p'로 들어갔었는데 2 번째 줄에서 댓글이 잘린다.

그래서 태그'a'에 있는 속성 onclick'으로 들어갔다.

```
ex_reple=ex_reple_html %>% html_nodes('.score_reple') %>% html_nodes('a') %>%
 html_attr('onclick')
head(ex_reple)
```

```
## [1] "javascript:showPointListByNid(15738216, 'after');parent.clickcr(this,
 'ara.uid', '', '', event); return false;"


## [2] "parent.clickcr(this, 'ara.report', '', '', event); common.report('fal
se','bril****', '8K2HlVSC7PqYju+2YgV+CK38AFbGMLQmwmH2e0ny9w8=', '비에 젖지 않
```

```
는 고급 장난감 텐트와, 비에 젖다 못해 잠기고 마는 반지하 가구 ', '15738216', 'poi
nt_after', false);return false;"
## [3] "javascript:showPointListByNid(15736627, 'after');parent.clickcr(this,
 'ara.uid', '', '', event); return false;"


## [4] "parent.clickcr(this, 'ara.report', '', '', event); common.report('fal
se','priv****', '37mh0NLZHsC3MNAXlIIrkJfakBHRUQCOfa34Z/i4Z18=', '최근 본 영화
중 가장 충격적이었음... 근데 보니까 15 세말고 19 세 걸어야될것같던데.. ', '15736627
', 'point_after', false);return false;"
## [5] "javascript:showPointListByNid(15737678, 'after');parent.clickcr(this,
 'ara.uid', '', '', event); return false;"


## [6] "parent.clickcr(this, 'ara.report', '', '', event); common.report('fal
se','papi****', 'elt+50xdwyLf3Nb05xt7o+jJsRMVUmRrui0UEwWJJd8=', '지하철이라는
단어가 언급되는 순간, 대다수의 관객은 자신이 어디에 이입할 지를 안다. ', '15737678
', 'point_after', false);return false;"
```

- **댓글이 있는 부분인 짝수행에서 댓글만 인덱싱 해준다.**

```
ex_reple_1=ex_reple[seq(2,length(ex_reple),2)]
ex_reple_2=str_sub(ex_reple_1,135,-52)
head(ex_reple_2)
```

```
## [1] "비에 젖지 않는 고급 장난감 텐트와, 비에 젖다 못해 잠기고 마는 반지하 가구"
## [2] "최근 본 영화중 가장 충격적이었음... 근데 보니까 15 세말고 19 세 걸어야될것
같던데.."
## [3] "지하철이라는 단어가 언급되는 순간, 대다수의 관객은 자신이 어디에 이입할 지
를 안다."
## [4] "전 가정부가 집 벨 누를 때 부터 이 영화는 장르가 바뀐다... 역대급 꿀잼영화"
## [5] "황금종려상 수상작을 자막 없이 볼 수 있다는 것 자체로 좋다."
## [6] "뭔가 모를 불쾌한 영화였다. 영화가 불쾌하다는게 아니라 보는 내내 가슴에 뭔가
영화에서 나오는 수석이 얹혀져있는 영화다."
```

- **댓글 추출**

```
movie_reple=list()
movie_reple_all=list()
for (i in 1:n){
  for (k in 1:m){
    reple_html_2=read_html(reple_url[[i]][k])
    m_reple=reple_html_2 %>% html_nodes('.score_reple') %>% html_nodes('a') %
```

```
>% html_attr('onclick')
    m_reple_1=m_reple[seq(2,length(m_reple),2)]
    movie_reple[[k]]=str_sub(m_reple_1,135,-52)
  }
  movie_reple_all[[i]]=unlist(movie_reple)
}
#각 리스트 이름을 영화 제목으로 바꾼다.
names(movie_reple_all)=movie_name
head(movie_reple_all[[1]],10)

##  [1] "비에 젖지 않는 고급 장난감 텐트와, 비에 젖다 못해 잠기고 마는 반지하 가구
"
##  [2] "최근 본 영화중 가장 충격적이었음... 근데 보니까 15 세말고 19 세 걸어야될것
같던데.."
##  [3] "지하철이라는 단어가 언급되는 순간, 대다수의 관객은 자신이 어디에 이입할 지
를 안다."
##  [4] "전 가정부가 집 벨 누를 때 부터 이 영화는 장르가 바뀐다... 역대급 꿀잼영화
"
##  [5] "황금종려상 수상작을 자막 없이 볼 수 있다는 것 자체로 좋다."
##  [6] "뭔가 모를 불쾌한 영화였다. 영화가 불쾌하다는게 아니라 보는 내내 가슴에 뭔
가 영화에서 나오는 수석이 얹혀져있는 영화다."
##  [7] "누군가의 냄새를 맡고, 평가하고, 묘사할 수 있는 것 또한 권력. 냄새로 서로
를 알아보고 경계하고 구분짓는 동물들의 세계와 우리 사회는 참 닮아있다."
##  [8] "등급조정이 필요해보입니다. 청소년들은 감당하기 버거운 내용이에요."
##  [9] "박서준이 잘못했네ㅋㅋㅋㅋ"
## [10] "누군가는 쏟아져 내리는 빗물을 장난감 텐트로도 막을 수 있지만 다른 누군가에
게는 똥구정물이 되어 차오른다. '계획'만으로는 올라갈 수 없는 우리네 사회구조의 냉혹
하고도 잔인한 현실이 세련된 연출력으로 수석처럼 무겁게 가슴을 누른다."
```

##6.시각화

필요한 패키지 설치

```
install.packages('KoNLP')
install.packages('wordcloud')
install.packages('RColorBrewer')
```

- **댓글에서 명사만 추출**

```
library(KoNLP)

## Warning: package 'KoNLP' was built under R version 3.5.3
```

```
## Checking user defined dictionary!

useNIADic()   #사용할 사전

## Backup was just finished!
## 983012 words dictionary was built.

word_1=movie_reple_all[1]
#댓글로부터 명사들만 추출
#USE.NAMES=T 이름 속성 반환 /
#USE.NAMES=F 이름 속성 없이 반환
word_2=sapply(word_1,extractNoun,USE.NAMES=F)
head(unlist(word_2),30)

##  [1] "비"     "고급"   "장난감" "텐트"   "비"     "반지하" "구"
##  [8] "영화"   "중"     "충격"   "적"     "15"     "세"     "19"
## [15] "세"     "것"     "데"     "지하철" "단어"   "언급"   "되"
## [22] "순간"   "대다수" "관객"   "자신"   "어디"   "이입"   "할"
## [29] "지"     "전"

word_3=unlist(word_2)
length(word_3)

## [1] 5048

#2 글자 이상 단어만 추출
word_3<-Filter(function(x){nchar(x)>=2},word_3)
length(word_3)

## [1] 3618

head(word_3,30)

##  [1] "고급"       "장난감"     "텐트"       "반지하"     "영화"
##  [6] "충격"       "15"         "19"         "지하철"     "단어"
## [11] "언급"       "순간"       "대다수"     "관객"       "자신"
## [16] "어디"       "이입"       "가정"       "영화"       "장르"
## [21] "역대급"     "꿀잼"       "황금종려상" "수상작"     "자막"
## [26] "자체"       "뭔가"       "불쾌"       "영화"       "영화"

wordcount=table(word_3)
head(sort(wordcount, decreasing=T),20)
```

```
## word_3
##   영화   사람   생각   기분 기생충   가난   냄새   봉준   가족     15
##    238     45     42     39     36     34     32     32     28     25
##   감독   장면   불쾌   들이 송강호   연기   재미   평점   현실    ^ㅋ
##     24     24     23     21     19     18     18     18     18     17
```

- **자음과 '영화' 제거**

```
word_4=str_replace_all(word_3,'[A-z ㄱ-ㅎ 0-9]','')
word_5=str_replace_all(word_4,'영화','')
```

- **워드 클라우드**

```
library(wordcloud)

## Warning: package 'wordcloud' was built under R version 3.5.3

## Loading required package: RColorBrewer

## Warning: package 'RColorBrewer' was built under R version 3.5.2

#워드클라우드작성
#freq : 빈도는 언급된 단어수
#scale : 폰트 사이즈 조정
#rot.per : 수직 텍스트의 비율 조정
#min.freq : 최소 몇번의 언급된 단어만 추출
#random.order : 순서 랜덤
#random.color : 컬러 랜덤
wordcloud(names(wordcount),freq=wordcount,scale=c(5,0.25),rot.per=0.25,min.fr
eq=5,random.order=F,random.color=T)
```

워드 클라우드