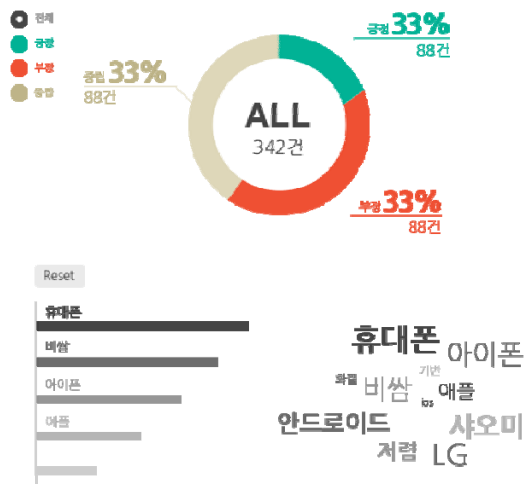
Wk16-3 : 웹문서 텍스트마이닝

- 한글 웹문서의 자연어 처리와 정보 추출 -

1. 텍스트 마이닝

- 텍스트 마이닝(text mining)이란, 다양한 알고리즘을 이용하여 대용량의 텍스트 문서로부터 트렌드와 관심어를 찾아내는 기법이다.

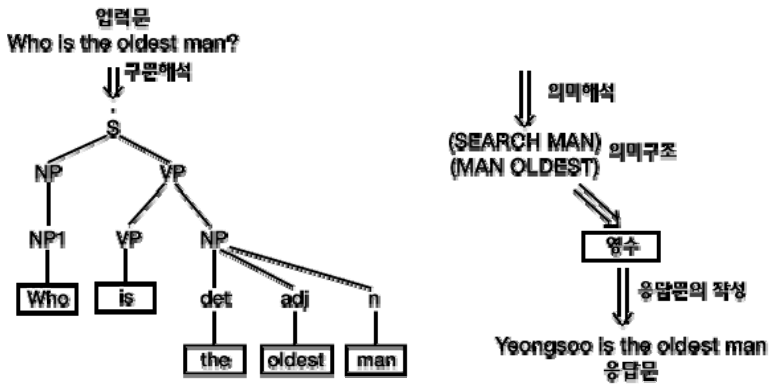


기법	설명
키워드 분석	단어 출현 빈도 추출 후 시각화
단어간 관계 분석	문서 내 출현 빈도 높은 단어 파악 후 단어간 상관관계 계산
감성 분석	긍정 단어와 부정 단어 빈도 추출 후 문서 감성 수치화

Image source: TREUM Co, Ltd.

2. 자연어 처리

- 자연어 처리(Natural Language Processing)란,
  컴퓨터로 사람 언어를 분석, 이해, 생성하는 기술을 일컫는다.



R 영어 자연어 처리 패키지
> install.packages("NLP")

R 한국어 자연어 처리 패키지
> install.packages("KoNLP")

Image source:『인공지능 입문 - 그림으로 풀어본』도우치 준이치

2. 자연어 처리

- 품사 분석: SimplePos09()

```
> t <- "즐겁게 텍스트 분석을 해봅시다!"

> SimplePos09(t)
$즐겁게
[1] " 즐겁/P+게/E "

$텍스트
[1] " 텍스트/N "

$분석을
[1] " 분석/N+을/J "

$해봅시다
[1] " 하/P+어/E+보/P+ㅂ시다/E "

$'!`
[1] " !/S "
```

태그 예시	품사
N	체언 (ex. 보통명사, 대명사, 수사)
P	용언 (ex. 동사, 형용사)
E	어미 (ex. 연결어미, 종결어미)
J	관계언 (ex. 격조사, 보조사)
S	기호
F	외국어

2. 자연어 처리

- str_match()를 이용한 N(체언), P(용언) 추출

```
> install.packages("stringr")

> library(stringr)

> tt <- SimplePos09(t)

> t_n <- str_match(tt, '([가-힣]+)/N')
> t_n
      [,1]          [,2]
[1,]  NA            NA
[2,]  "활용/N"      "활용"
[3,]  NA            NA
[4,]  "텍스트/N"    "텍스트"
[5,]  "분석/N"      "분석"
[6,]  NA            NA
[7,]  NA            NA
```

```
> t_p <- str_match(tt, '([가-힣]+)/P')
> t_p
      [,1]          [,2]
[1,]  NA            NA
[2,]  NA            NA
[3,]  "즐겁/P"      "즐겁"
[4,]  NA            NA
[5,]  NA            NA
[6,]  "하/P"        "하"
[7,]  NA            NA
```

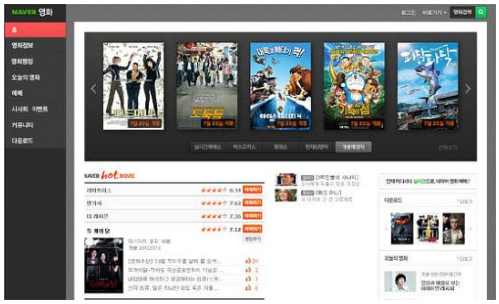

3. 워드 클라우드

- 워드 클라우드(word cloud)는 텍스트의 키워드, 개념을 직관적으로 파악하도록 핵심 단어를 시각적으로 보여주는 기법이다.

Image source: Google Image

4. 웹문서의 텍스트마이닝 실습

1) '네이버 영화'에서 영화 <머니볼>의 네티즌 140자 평을 '크롤링'한다.
2) '단어-문서 행렬'을 산출한다.
3) 출현 빈도가 높은 단어들의 '워드 클라우드'를 생성한다.
4) 단어 사이의 상관관계를 보여주는 '동시 발생 행렬'을 산출한다.
5) 동시 발생 행렬을 바탕으로 단어들의 '네트워크'를 생성한다.



16-3. 웹문서 텍스트마이닝

4. 웹문서의 텍스트마이닝 실습

- 패키지 설치

```
install.packages("xml2")          # html 처리
install.packages("rvest")         # html 처리
install.packages("KoNLP")         # 국문 자연어 처리
install.packages("tm")            # 정보 추출 도구
install.packages("stringr")       # string 처리
install.packages("wordcloud")     # 워드 클라우드
# install.packages("qgraph")      # 네트워크
```

4. 웹문서의 텍스트마이닝 실습

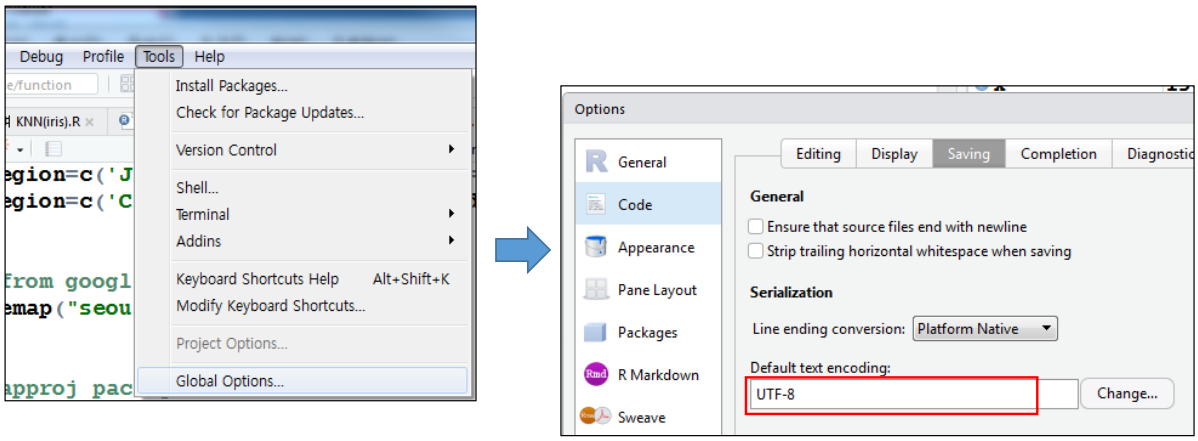
- 패키지 설치

```
# lec16_3_tm.R
# Crawling
# textmining

## modify and use the code right below only if there is a problem
# Sys.setenv(JAVA_HOME='C:\\Program Files\\Java\\jre1.8.0_151')

# install packages
install.packages("xml2") # to read html
install.packages("rvest") # to use 'html_nodes'
install.packages("KoNLP") # korean natural language processing
install.packages("tm") # corpus, term-document matrix, etc.
install.packages("stringr") # to use 'str_match'
install.packages("wordcloud") # word cloud
# install.packages("qgraph") # qgraph of co-occurence matrix
```


4. 웹문서의 텍스트마이닝 실습

- 인코딩(UTF-8)방식 변경 (한글 인코딩)

Tools_Global options 메뉴에서 Code_Saving으로 가서 UTF-8로 변경



4. 웹문서의 텍스트마이닝 실습

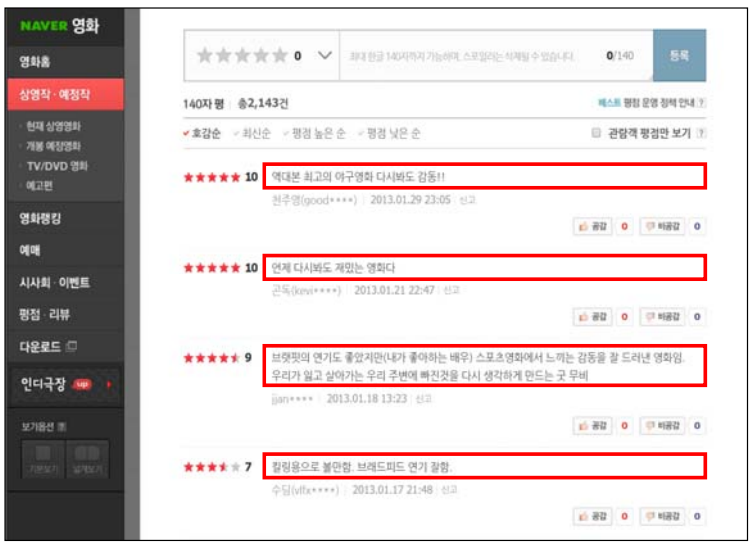
- **크롤링(crawling)**

  - 웹페이지를 그대로 가져와서 데이터를 추출해 내는 행위이다.
  - 데이터를 개인 하드에 소장하는 것은 합법이나, 배포는 불법이다.

16-3. 웹문서 텍스트마이닝

4. 웹문서의 텍스트마이닝 실습

- **네이버 영화 <머니볼> 네티즌 140자 평**
  http://movie.naver.com/movie/bi/mi/point.nhn?code=51786



마우스 오른쪽 클릭 후,
페이지 소스 보기

또는

ctrl + U !

4. 웹문서의 텍스트마이닝 실습

```
url_base <-  # 크롤링 대상 URL
'http://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=51786&type=after&isActualPointWrite
Execute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&page='


reviews <- c()  # 140자 평 저장 벡터


for(page in 1:10){
  url <- paste(url_base, page, sep='')                    # 1~10페이지까지 페이지 수집
  htxt <- read_html(url)
  comment <- html_nodes(htxt, 'div')%>%html_nodes('div.input_netizen')%>%
    html_nodes('div.score_result')%>%html_nodes('ul')%>%html_nodes('li')%>%
    html_nodes('div.score_reple')%>%html_nodes('p')    # 140자 평 위치
  review <- html_text(comment)                            # 실제 리뷰의 text 파일만 추출
  review <- repair_encoding(review, from = 'utf-8')              # 인코딩 변경
  review <- str_trim(review)                              # 앞뒤 공백 문자 제거
  reviews <- c(reviews, review)                           # 결과값 저장
}
```


6. 자연어 처리

- 용언, 체언 추출 function

```
ext_func <- function(doc){

   doc_char <- as.character(doc)

   ext1 <- paste(SimplePos09(doc_char))                  # 품사 분석

   ext2 <- str_match(ext1, '([A-Z가-힣]+)/[NP]')         # 용언, 체언 선택
   keyword <- ext2[,2]

   keyword[!is.na(keyword)]                              # NA 값 제외

}
```

7. 정보 추출

- 단어-문서 행렬

```
corp <- Corpus(VectorSource(reviews))  # 말뭉치(corpus) 생성

tdm <- TermDocumentMatrix(corp,  # 용어-문서 행렬 산출
                          control=list(
                            tokenize=ext_func,
                            removePunctuation=T,
                            removeNumbers=T,
                            wordLengths=c(4,8)))

tdm_matrix <- as.matrix(tdm)  # 행렬로 변환
Encoding(rownames(tdm_matrix)) <- "UTF-8"  # 인코딩

word_count <- rowSums(tdm_matrix)  # 각 단어별 총 출현 빈도 계산
word_order <- word_count[order(word_count, decreasing=T)]  # 내림차순 정렬

doc <- as.data.frame(word_order)  # 데이터 프레임으로 변환
```

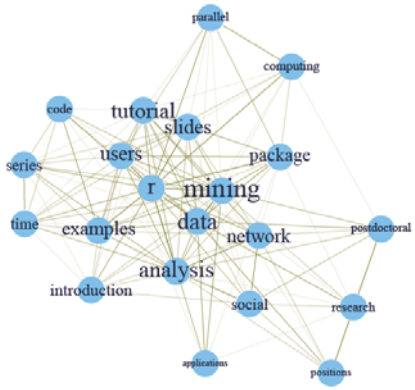

8. 워드 클라우드 구현

- wordcloud() 함수

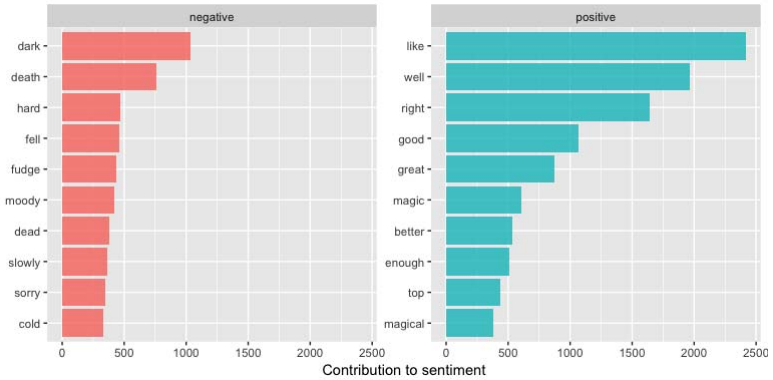
```
windowsFonts(font=windowsFont("맑은 고딕"))

set.seed(1234)  # 동일한 워드 클라우드 생성 (난수 고정)

wordcloud(names(wordcount),       # 키워드
          freq=wordcount,         # 빈도
          min.freq=2,             # 최소 출현 빈도
          max.words=30,           # 출력 키워드 수
          random.order=FALSE,     # 고빈도 키워드 중앙 배치
          scale=c(5,1),           # 키워드 크기 범위
          rot.per=0.35,           # 회전 키워드 비율
          family="font", colors=brewer.pal(8,"Dark2"))
```

9. 고급 텍스트 마이닝

• 단어간 관계분석



• 감성분석



An Example of Social Network Analysis with R using Package igraph
https://www.r-bloggers.com/an-example-of-social-network-analysis-with-r-using-package-igraph/

Text Mining: Sentiment Analysis
http://uc-r.github.io/sentiment_analysis