

[11주차]

# 비정형 텍스트 분석: 전처리와 키워드 분석

---

# 1. 텍스트 분석 기법들

---

# 텍스트 분석기법들

---

- 자연어 처리/전처리
- 토픽모델링
- 텍스트 크러스터링
- 감성분석
- 소셜 네트워크 분석 등

# 자연어 처리/전처리

---

- 텍스트 분석을 위한 전단계 작업으로
- 자연어(natural language)로 표현된 텍스트 데이터에서 컴퓨터가 이해할 수 있도록 단어들을 식별하고
- 도메인에 적합한 의미 정보로 변환하여 대표적인 단어를 추출하는 것
- 형태소 분석, 불용어 처리, 어간처리, 가중치와 Term-Document Matrix도출 등

# 자연어 처리/전처리

## ■ 텍스트 전처리의 예

### Unstructured Document

Doc 1 : deposit the cash and check in the bank!!  
Doc 2 : the river boat is on the bank  
Doc 3 : borrow based on credit  
Doc 4 : river boat floats up the river  
Doc 5 : boat is by the dock near the bank

### Structured Matrix

#### Term-by-Document Frequency Matrix

	d1	d2	d3	d4	d5
cash	1	0	0	0	0
check	1	0	0	0	0
bank	1	1	0	0	1
river	0	1	0	2	0
credit	0	0	1	0	0

Pre-processing

# 토픽(Topic) 모델링

---

- 텍스트 문서 집합 내에 잠재된 주제를 도출하는 분석 기법
- 개별 문서는 다수의 주제, 혹은 토픽을 다룰 수 있다 는 점을 가정
- 수집된 텍스트를 토픽의 확률적 혼합체로 간주하고, 각 토픽을 추출된 키워드들의 분포로 나타냄으로써 텍스트 내의 구조를 파악
- 뉴스 등 대용량의 텍스트 데이터로부터 주요 이슈 들을 도출해 내는데 활용

# 텍스트 클러스터링

---

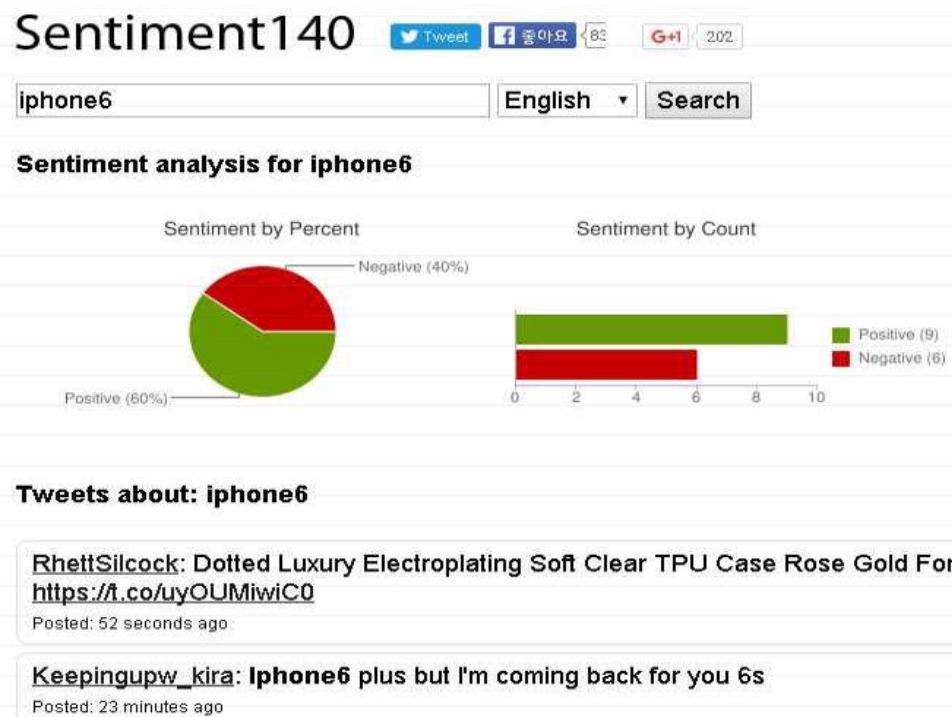
- 텍스트 문서에 군집분석을 적용하는 것
- 문서별로 도출되는 주요 단어를 중심으로  
유사 문서군을 도출하고
- 각 집단의 성격을 파악함으로써  
텍스트 전체 구조를 이해하는 탐색적 분석기법
- 계층적 클러스터링과 비계층적 클러스터링으로 구분됨

# 감성분석

---

- 사람들이 작성한 텍스트를 분석하여  
특정 주제에 대해 의견이 긍정, 부정,  
또는 중립인지를 분류하는 방법
- 트위터나 페이스북과 같은 SNS 및 인터넷 댓글 등  
특정 사안에 대한 오피니언(Opinion)이 잠재되어 있는  
문서들을 주로 분석
- 오피니언 마이닝(opinion mining)이라고도 불림

# 감성분석



# 소셜 네트워크 분석

---

- 개인과 집단들 간의 사회적 관계를 노드와 링크로서 구조적으로 분석하여 내재된 관계를 파악하는 기법
- 구성원들 간의 연결 구조 및 연결 강도 등을 정량적인 방법으로 분석함으로써 내재된 현상을 찾아냄

## 2. 텍스트의 전처리와 키워드 분석

---

## 2.1 텍스트의 전처리

---

# 텍스트 전처리(Preprocessing)란?

---

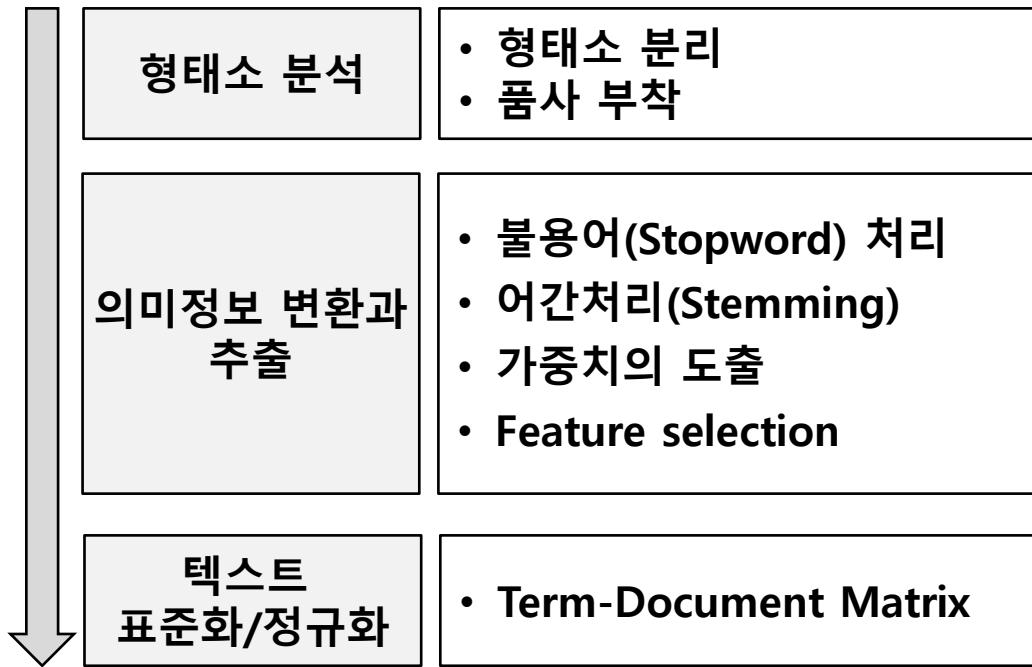
## ■ 텍스트 전처리란?

- 텍스트 분석을 위한 전 단계 작업으로
- 자연어(natural language)로 표현된 텍스트 데이터에서  
컴퓨터가 이해할 수 있도록 단어들을 식별하고
- 도메인에 적합한 의미 정보로 변환하여 대표적인 단어를 추출하는 것
- 이를 표준화, 또는 정규화하는 과정을 포함

# 텍스트 전처리(Preprocessing)란?

---

## ■ 주요 과업



# 미리보기

## ■ 텍스트 전처리의 예

### Unstructured Document

Doc 1 : deposit the cash and check in the bank!!  
Doc 2 : the river boat is on the bank  
Doc 3 : borrow based on credit  
Doc 4 : river boat floats up the river  
Doc 5 : boat is by the dock near the bank

### Structured Matrix

#### Term-by-Document Frequency Matrix

	d1	d2	d3	d4	d5
cash	1	0	0	0	0
check	1	0	0	0	0
bank	1	1	0	0	1
river	0	1	0	2	0
credit	0	0	1	0	0

Pre-processing

# 형태소 분석이란?

---

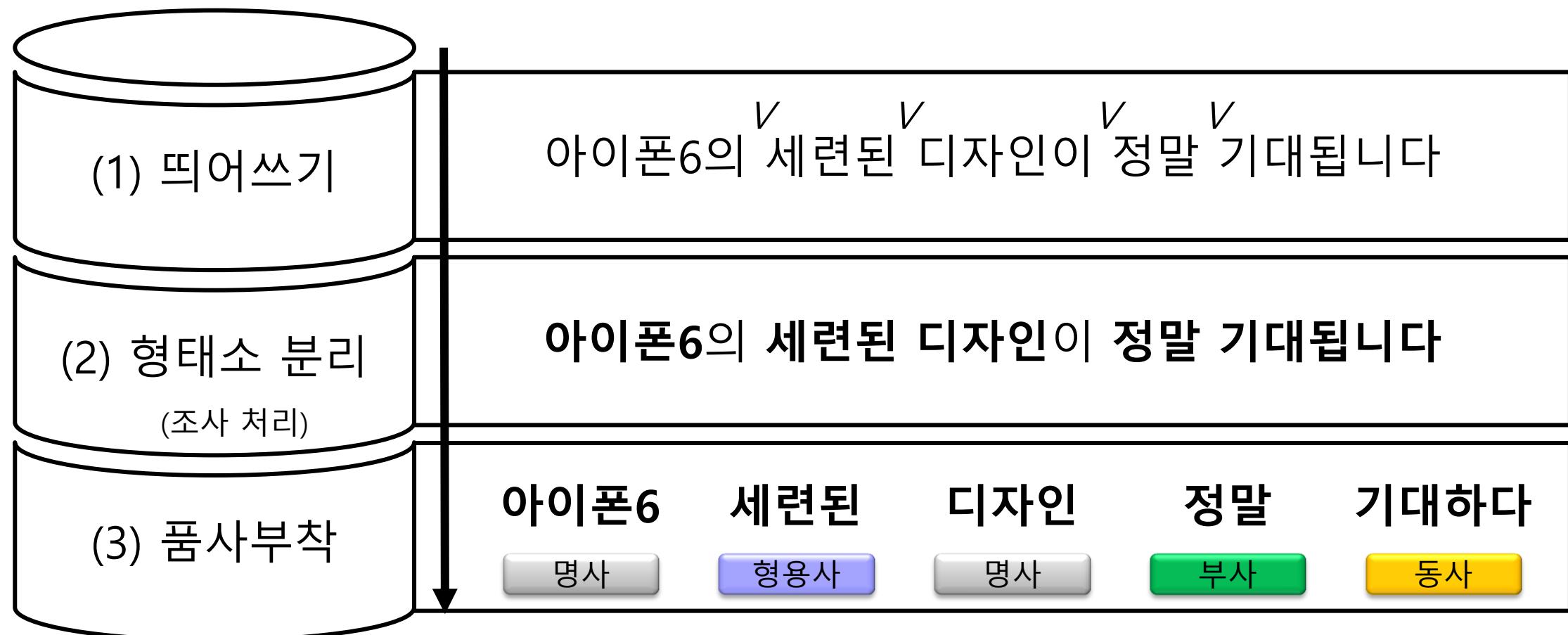
- 형태소(morpheme)란?

- 의미를 가지는 가장 최소의 단위

- 형태소 분석 단계에서는...

- 단어(또는 어절)를 구성하는 각 형태소를 분리
  - 각 단어에 대한 품사 부착 수행
  - 구두점, 숫자 등 의미를 가지지 않는 요소들을 제거하고,  
오탈자, 띄어쓰기 등 기본적인 텍스트에 대한 기본적인 처리 수행

# 형태소 분석 예시



# 의미정보 변환과 추출이란?

---

- 의미 있는 형태소 정보를 선별하여 저장하는 단계
- 텍스트 데이터를 검토하여 정보나 지식을 추출 할 수 있는 기반을 잡아가는 과정
- 세부 내용
  - 불용어 (Stopword) 처리
  - 어간처리 (Stemming)
  - 가중치의 도출
  - 특성(용어)추출(Feature selection)

# 불용어 (Stopword) 처리

---

## ■ 불용어 처리란?

- 제거할 용어를 지정하는 작업을 말함

## ■ 불용어 처리 방법

- 형태소 분리된 데이터를 처리자가 보면서 판단하여 제거하는 방법
  - 의미 없는 단어, 사용하지 않는 단어 등 제거
- 미리 정해진 단어 리스트에 대조하여 제거하는 방법
- 기타 자동으로 제거하는 여러 가지 방법이 개발되어 있음

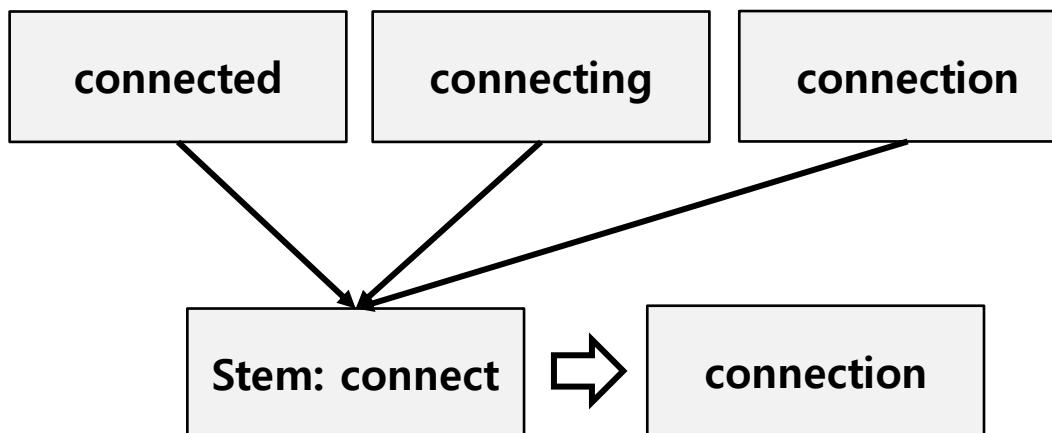
# 어간 처리 (Stemming)

---

## ■ 어간 처리란?

- 어간 (Stem) 파악을 통해서 동일한 어간을 가지고 있는 단어는 하나의 단어로 처리하는 작업
- 같은 어근을 가진 다른 용어를 같게 취급할지 다르게 취급할지를 정함

## ■ 어간 처리의 예



# 중요도의 산출: TF-IDF란?

---

- Term Frequency-Inverse Document Frequency의 약자로  
“단어 빈도”-“역문서 빈도”를 의미
- 여러 문서의 집합에서 특정 단어가 얼마나 중요한가를 판단할 수 있는 값
- 정보 추출(Information Retrieval) 분야에서 단어별 가중치 산출을 위해 범용적으로 사용되고 있음 (Salton & McGill, 1983)
- 생각해 봅시다!
  - 어떤 단어가 자주 나오면?
  - 어떤 단어가 거의 안 나오면?
  - 어떤 단어가 여러 문서에서 나오면?

# 중요도의 산출: TF-IDF란?

---

## ■ TF (단어 빈도, Term Frequency)

- 특정 키워드가 문서 내에 얼마나 자주 등장하는지를 나타내는 값
- 이 값이 클수록 문서에서 중요한 키워드라고 볼 수 있음

## ■ DF (문서 빈도, Document Frequency)

- 특정 단어가 문서 집합 내에서 얼마나 공통으로 출현하는지를 나타내는 값
- 어떤 단어가 문서 집합 내에서 빈번하게 사용되는 것은 그 단어가 흔하다는 것을 의미하기도 함

## ■ IDF (문서 빈도의 역수, Inverse Document Frequency)

# 중요도의 산출: TF-IDF란?

---

## ■ 산식

$$\text{TF-IDF} = \text{TF} \times \log(n/\text{DF})$$

- TF = 문서 내 특정 키워드의 빈도수
- n = 전체 문서 빈도수
- DF = 단어가 출현한 문서의 빈도수
- IDF = DF의 역수

# 단어-문서 행렬(Term-Document Matrix)

---

- 전처리 과정을 통하여 “비구조적인” 문서 집합은 “구조화”된 단어-문서 행렬(Term-Document Matrix)로 변환
- m개의 단어와 n개의 문서로 이루어진 행렬,  $d_{ij}$ 는 i단어와 j문서에 포함되어 있는지의 여부나 빈도수를 의미
- 단어와 문서 간 관계를 나타내기 위해 TF-IDF 가중치를 사용하기도 함

	문서1	문서2	문서3	…	문서n
단어1	$d_{11}$	$d_{12}$	$d_{13}$	…	$d_{1n}$
단어2	$d_{21}$	$d_{22}$	$d_{23}$	…	$d_{2n}$
단어3	$d_{31}$	$d_{32}$	$d_{33}$	…	$d_{3n}$
⋮	⋮	⋮	⋮		⋮
단어m	$d_{m1}$	$d_{m2}$	$d_{m3}$	…	$d_{mn}$

# 다시 보기

## ■ 텍스트 전처리의 예

### Unstructured Document

Doc 1 : deposit the cash and check in the bank!!  
Doc 2 : the river boat is on the bank  
Doc 3 : borrow based on credit  
Doc 4 : river boat floats up the river  
Doc 5 : boat is by the dock near the bank

### Structured Matrix

#### Term-by-Document Frequency Matrix

	d1	d2	d3	d4	d5
cash	1	0	0	0	0
check	1	0	0	0	0
bank	1	1	0	0	1
river	0	1	0	2	0
credit	0	0	1	0	0

Pre-processing

# 텍스트의 표준화/정규화

---

- 문서의 길이가 긴 문서일수록...
  - 추출될 수 있는 단어의 수가 많고,
  - 각 단어의 출현 빈도 또한 높아서
  - 길이가 짧은 문서에 비해 중요하다고 간주될 가능성이 높음
- 문서의 길이에 대한 영향력을 감소시키기 위해...
  - TF에 대한 표준화(standardization) 또는 정규화(normalization) 수행

# 텍스트의 표준화/정규화

---

## ■ 표준화 TF 산식

- X를 TF라고 하면 표준화 변수 z는 다음과 같음

$$z = (x - \bar{x})/s$$

x: 각 문서에서의 평균

s: 표준편차

## ■ 정규화 TF 산식

- X를 TF라고 하면 정규화 변수 z는 다음과 같음

$$Z = (1 + \log_2 X) / n_i$$

TF=0이면 위 식의 값은 0으로 정의

n<sub>i</sub> : 문서i 에서 총 단어의 빈도수

# 텍스트의 표준화/정규화

## ■ Term-Document Matrix

	TF			정규화 TF-IDF		
	doc 1	doc 2	doc 3	doc 1	doc 2	doc 3
Aaa	2	2	2	0 (=0.4*log <sub>2</sub> (3/3))	0 (=0.4*log <sub>2</sub> (3/3))	0 (=0.4*log <sub>2</sub> (3/3))
Bbb	1	0	1	0.117 (=0.2*log <sub>2</sub> (3/2))	0	0.117 (=0.2*log <sub>2</sub> (3/2))
Ccc	1	1	1	0 (=0.2*log <sub>2</sub> (3/3))	0 (=0.2*log <sub>2</sub> (3/3))	0 (=0.2*log <sub>2</sub> (3/3))
Ddd	1	0	0	0.317 (=0.2*log <sub>2</sub> (3/1))	0	0
Eee	0	0	1	0	0	0.317 (=0.2*log <sub>2</sub> (3/1))
Fff	0	1	0	0	0.317 (=0.2*log <sub>2</sub> (3/1))	0
Ggg	0	1	0	0	0.317 (=0.2*log <sub>2</sub> (3/1))	0

# 정리하기

## ■ 텍스트 전처리의 예

### Unstructured Document

Doc 1 : deposit the cash and check in the bank!!  
Doc 2 : the river boat is on the bank  
Doc 3 : borrow based on credit  
Doc 4 : river boat floats up the river  
Doc 5 : boat is by the dock near the bank

### Structured Matrix

#### Term-by-Document Frequency Matrix

	d1	d2	d3	d4	d5
cash	1	0	0	0	0
check	1	0	0	0	0
bank	1	1	0	0	1
river	0	1	0	2	0
credit	0	0	1	0	0

Pre-processing

## 2.2 키워드 분석과 워드 크라우드

---

# 키워드 분석이란?

---

- 문서의 주제를 알기 위해 원본 텍스트 내의 단어를 분석하는 통계적인 방법
- 텍스트에서 사용된 단어들의 출연 빈도에 따라 의미 또는 트렌드를 파악하는 분석 방법
- 빈도에 기초한 가장 기초적인 텍스트 분석 기법

# 키워드 분석 방법

---



# 예시\_워드 크라우드

---

- 기본적인 텍스트 데이터의 시각화 방법론
- 텍스트에 포함된 주요 단어나 어절의 중요도를 폰트의 사이즈나 색으로 표현
- 문서 상 가장 중요한 용어들에 대한 인식 용이

# 예시\_워드 크라우드



- 빅데이터란 무엇인가?
  - 연관된 주요 요소들은?
  - 빅데이터로 할 수 있는 일은?
  - 관련하여 제기되는 이슈들은?

## 2.3 R을 활용한 워드클라우드 실습

---

# 키워드 분석 – 실습 예제

## ■ ‘곡성’ 영화평에 대한 키워드 분석을 수행하고자 한다.

(데이터: 개봉 전 150개 / 개봉 후 150개의 게시글)

★★★★★ 1 정신적으로 너무 안좋다  
박시영(tldu\*\*\*\*) | 2016.05.11 20:18 | 신고  
공감 5598 비공감 2600

★★★★★ 1 관람객 ... 할말없는 영화임. 본 사람은 공감할듯.. 무섭고 잔인하고 징그럽고 소름끼치는 반전 까지 고루 갖췄지만 그 모든걸 넘어서는 찜찜함.  
정연비(baby\*\*\*\*) | 2016.05.11 22:05 | 신고  
공감 6387 비공감 3590

★★★★★ 10 오늘 개봉 오후 5시 뷰터인데 뭘 조조부터 보고왔대 조조 영화시간표에 곡성 없어 지금 보고왔다고 하는놈을 알바임 알바도 쓰려면 지능이 높아야지 서울 및 수도권 영화 개봉시간이 오후 5시인데 보고왔다고 구라까면 되나  
Kondre(osh6\*\*\*\*) | 2016.05.11 09:05 | 신고  
공감 3686 비공감 1225

★★★★★ 8 관람객 보는 내내 골름 생각났음ㅋㅋ  
psy5\*\*\*\* | 2016.05.11 20:53 | 신고  
공감 3100 비공감 715

★★★★★ 9 관람객 맨정신으로 작두에 올라 영화보는 기분  
설난황소(gust\*\*\*\*) | 2016.05.11 20:42 | 신고  
공감 2400 비공감 419

A
1 text
2 드디어 나홍진 감독의 영화가 나오는구나
3 향해, 주격자 두개다본 사람으로써 나홍진 감독 기대합니다
4 솔직히 황해 개지렸는데 너무 과소평가 받는 듯 —— 곡성 기대합니다 ^^
5 언제개봉하던 믿고 봅니다
6 나홍진 감독이면 기대 평점 10은 되야겠죠 ^^
7 나홍진 감독에 광도원+황정민... 무슨 말이 더 필요한가
8 천우희 배우님 짱이에요!!! 기대할께요!!^^이번에도 멋진연기 보여주시길
9 정말 기다리고 기다리던 나홍진 감독의 영화 기대합니다 ㅠㅠ김윤석 하정우 캐스팅 한번더?
10 드디어!! 많이 기다렸습니다 천우희 유일한 여주! 기대기대합니다 곡성 화이팅 대박터트립시당?
11 나홍진이라는 이름만으로 신뢰도가 최상
12 난 ~ 전남 곡성 출신!!
13 나홍진.. 다시한번 미쳐보자
14 을 해 꼭 볼 수 있기를 . 황해만한 웰리티를 지닌 한국영화를 아직 보지 못하네요.
15 광도원, 천우희 정말 좋아하는 배우들만 나오네
16 나홍진영화는 정말기대된다
17 추격자, 황해에 이어 곡성... 기대된다 나홍진감독의 작품 빨리보고싶다
18 기다립니다. 무조건 십점만점
19 도대체 언제 나옵니까? 보고싶어 미칠거 같습니다..
20 드디어 나홍진 감독의 신작이!!
21 대작이 또 한번 나오는군요

<수집된 데이터 예시>

# 키워드 분석 – 실습 예제...

```
1 install.packages("rJava")
2 install.packages("KoNLP")
3 install.packages("stringr")
4 install.packages("wordcloud")
5 install.packages("RColorBrewer")
6
7 library(rJava)
8 library(KoNLP)
9 library(stringr)
10 library(wordcloud)
11 library(RColorBrewer)
12
13 before <- read.csv("before.csv", stringsAsFactors=F)
14
15 useSejongDic()
16
17 mergeUserDic(data.frame(c('곽도원', '황정민', '쿠니무라', '나홍진'), "ncn"))
18 mergeUserDic(data.frame('곡성', "ncn"))
```

① 키워드 분석과 시각화 (워드 클라우드)를 구현하기 위해 필요한 패키지를 설치하고 라이브러리 호출한다.

② 분석에 필요한 데이터 불러오기

③ 한글을 분석하기 위한 세종 사전 불러오기

④ 한글사전에 연구자가 필요로 하는 단어를 추가 할 수 있다. 2개 이상의 단어를 추가 할 때 c로 묶어서 추가 가능하다

# 키워드 분석 – 실습 예제...

```
19 word <- sapply(before$text, extractNoun, USE.NAMES = F)  
20 word <- unlist(word)
```

- ⑤ 불러온 파일에서 text변수에 있는 내용으로  
부터 단어를 추출한다. apply함수를 단순화  
한 sapply()함수(simplify apply)를 사용

```
21 word1 <- Filter(function(x){nchar(x)>=2&nchar(x)<=5},word)  
22 head(word1,20)  
23  
24 wordcount <-table(unlist(word1))  
25 head(sort(wordcount, decreasing = T),50)
```

- ⑥ 두 글자부터 다섯 글자의 단어를 추출해서  
새로운 데이터로 저장하고 head()함수로 확  
인한다. 즉, 한 글자의 단어나, 여섯 글자 이  
상의 단어는 키워드로 보기 어렵다고 가정  
한다.

```
27 word1 <- gsub("\n","",word1) #엔터 표시 제거  
28 word1 <- gsub("\d+","",word1) #숫자 제거  
29 word1 <- gsub("\.",","",word1) #마침표 제거  
30 word1 <- gsub("곽도원님","곽도원",word1)  
31 word1 <- gsub("정말","",word1)  
32 word1 <- gsub("중이","",word1)  
33 word1 <- gsub("하게","",word1)  
34 word1 <- gsub("이거","",word1)  
35 word1 <- gsub("8.5","",word1)  
36 word1 <- gsub("대하","",word1)
```

- ⑦ gsub("")함수를 사용하여 필요 없는 단어를  
제거하거나 변경한다.  
 $X \leftarrow \text{gsub("제거할 단어", "변경할 단어", X)}$

# 키워드 분석 – 실습 예제...

```
38 write(unlist(word1), "word.txt")
39 wordlist <- read.table("word.txt")
40 nrow(wordlist)
```

⑧ 1차 정제된 파일을 txt 형태로 저장 가능하다.

```
42 wordcount2 <- table(wordlist)
43 write.csv(wordcount2, "frequency.csv")
44 head(sort(wordcount2, decreasing = T), 50)
45
46 palete<-brewer.pal(6, "Dark2")
47 windowsFonts(malgun=windowsFont("맑은 고딕"))
```

⑨ 생성할 워드 클라우드를 구성하는 단어에 대한 색과 폰트를 설정한다.

```
49 wordcloud(names(wordcount2), freq=wordcount2,
50   scale=c(5,1), rot.per=0.5, min.freq = 2,
51   random.order = F, random.color = T,
52   colors=palete, family="malgun")
```

⑩ 워드 클라우드 그리기  
#scale = 출력되는 단어 간의 크기 비율  
#rot.per = 단어들 간 간격 조정  
#min.freq = 텍스트 문서 집합 내 단어 출현 횟수

# 키워드 분석 - 실습 예제...

