



서강대학교 정보통신 대학원 정 화민 교수 (MIS Ph.D.)

* 텍스트 마이닝

- 문자로 된 데이터에서 가치 있는 정보를 얻어내는 분석 기법을 Text Mining이라고 한다.
- 문장을 구성하는 어절들이 어떤 품사로 되어 있는지 파악하는 “형태소 분석(Morphology Analysis).
- 형태소 분석으로 어절들의 품사를 파악한 후 명사, 동사 형용사 등 의미를 지닌 품사의 단어들을 추출해 각 단어가 얼마나 많이 등장했는지 확인함.
- 텍스트 마이닝을 통해 SNS나 웹 사이트에 올라온 글을 분석하면 사람들이 어떤 이야기를 나누고 있는지 파악할 수 있다.
- R 프로그래밍에서 한글 자연어 분석 패키지 KoNLP를 이용하면 한글 데이터 형태소 분석을 할 수 있다.
(Java가 설치되어 있어야 함)
- 필요 패키지 : rJava, memoise, KoNLP

텍스트 마이닝 스크립트

```
# 패키지 설치
install.packages("rJava")
install.packages("memoise")
install.packages("KoNLP")
install.packages("dplyr")

# 패키지 로드
library(KoNLP)
library(dplyr)

# java 폴더 경로 설정
Sys.setenv(JAVA_HOME="C:/Program Files/Java/jre1.8.0_111/")

useNIADic()

# 데이터 불러오기
txt <- readLines("hiphop.txt")
head(txt)

install.packages("stringr")
library(stringr)

# 특수문제 제거
txt <- str_replace_all(txt, "\\w", " ")

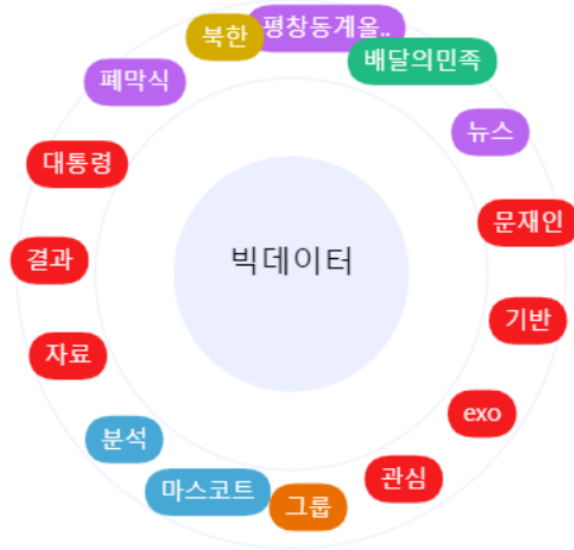
참조: R 데이터 분석
```

시각화

* <http://www.socialmetrics.co.kr/>를 통한 SNS 분석

빅데이터 연관어 맵 ?

(기간 : 2018.10.30 ~ 2018.11.29)



- | | | |
|-------|---------|---------|
| ✓ 인물 | ✓ 단체 | ✓ 장소 |
| ✓ 상품 | ✓ 브랜드 | ✓ 시사/경제 |
| ✓ 라이프 | ✓ 문화/여가 | ✓ 속성 |

빅데이터 연관어 언급량 ?

(기간 : 2018.10.30 ~ 2018.11.29)



분석	35827
평창동계올림픽	32546
exo	32397
결과	30689
뉴스	12263
그룹	5098
폐막식	3654
기반	3576
자료	3412
배달의민족	3398
관심	2101
대통령	1770
문재인	1761
마스크트	1760
북한	1743

시각화

* 빅카인즈: <https://www.kinds.or.kr/>를 통한 분석

뉴스빅데이터 분석시스템, 뉴스 속 키워드 관계망, 주요 이슈, 정보원, 이슈 트렌드 분석, 정보제공. [고신문](#) · [분석사례](#) · [최신뉴스](#) · [이슈모음](#)

시각화 분석 서비스

검색한 키워드와 관련한 뉴스를 분석해 다양한 시각화 차트를 제작할 수 있습니다.
시각화 차트는 다른 이용자에게 공유할 수 있습니다.



키워드트렌드

검색어와 관련된 기사의 수를 시기별 그래프로 보여줍니다.



연관어분석

검색어와 연관된 키워드를 워드클라우드로 보여줍니다.



네트워크분석

오늘이슈, 오늘인물, 분야키워드에서 선택한 항목과 관계된 개체(인물, 기관, 장소 등)를 네트워크 형태로 보여줍니다.



공공데이터융합

뉴스 분석 데이터와 주가지수, 인구통계, 등과 같은 공공데이터를 융합하여 그래프로 보여줍니다.

* Word Cloud

- 단어의 빈도를 구름모양으로 표현한 그래프.
- 워드클라우드를 만들면 단어의 빈도에 따라 글자의 크기와 색깔이 다르게 표현됨
- R 프로그래밍 패키지: word cloud , 글자 색깔을 표현하는 RColorBrewer 사용 (R에 내장)

Word cloud 스크립트

```
# 패키지 설치
install.packages("wordcloud")

# 패키지 로드
library(wordcloud)
library(RColorBrewer)

pal <- brewer.pal(8,"Dark2") # Dark2 색상 목록에서 8개 색상 추출

set.seed(1234)
wordcloud(words = df_word$word, # 단어
          freq = df_word$freq, # 빈도
          min.freq = 2,        # 최소 단어 빈도
          max.words = 200,     # 표현 단어 수
          random.order = F,    # 고빈도 단어 중앙 배치
          rot.per = .1,        # 회전 단어 비율
          scale = c(4, 0.3),   # 단어 크기 범위
          colors = pal)        # 색깔 목록

pal <- brewer.pal(9,"Blues")[5:9] # 색상 목록 생성
set.seed(1234)                    # 난수 고정

wordcloud(words = df_word$word, # 단어
          freq = df_word$freq, # 빈도
          min.freq = 2,        # 최소 단어 빈도
          max.words = 200,     # 표현 단어 수
          random.order = F,    # 고빈도 단어 중앙 배치
          rot.per = .1,        # 회전 단어 비율
          scale = c(4, 0.3),   # 단어 크기 범위
          colors = pal)        # 색상 목록
```

참조: R 데이터 분석

* Word Cloud Generator를 활용한 Word Cloud
<https://www.iasondavies.com/wordcloud/>



Paste your text below!

How the Word Cloud Generator Works

The layout algorithm for positioning words without overlap is available on GitHub under an open source license as d3-cloud. Note that this is the only the layout algorithm and any code for converting text into words and rendering the final output requires additional development.

As word placement can be quite slow for more than a few hundred words, the layout algorithm can be run asynchronously, with a configurable time step size. This makes it possible to animate words as they are placed without stuttering. It is recommended to always use a time step

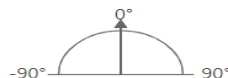
Go!

Spiral: ☒ Archimedean ☐ Rectangular

Scale: ☒ $\log n$ ☐ \sqrt{n} ☐ n

Font: Impact

5 orientations from 0 ° to 90 °



Number of words: 250

☐ One word per line

Download: [SVG](#)