사례연구 5

텍스트 마이닝

조 : B3 조
조장 : 유준
조원 : 문현진, 임성현
작성일자 : 21.11.25

목 차

1. 서론

이 보고서의 목적은 앞서 학습한 텍스트 데이터 분석을 활용하여 사례연구 주제에 대한 분석을 실시하고 그에 따른 결과를 알아보는 것에 있다.

□ 사례연구 주제

① 링컨 대통령 연설문을 대상으로 토픽분석, 단어구름 시각화, 연관어 분석, 연관어를 시각화 및 설명

② 다음(Daum.net) 포털 사이트의 실시간 뉴스를 대상으로 토픽분석, 단어구름 시각화, 분석시점 주요 이슈 확인

2. 본론

1) 링컨 대통령 국문 번역 연설문 텍스트 데이터 분석

(1) 링컨 대통령 국문 번역 연설문

링컨 게티스버그 연설문 (1863.11.19)
87년 전 우리의 선조들은 이 대륙에 자유의 정신으로 잉태되고 만인이 평등하게 창조되었다는 신념이 바쳐진 새로운 나라를 세웠습니다.
지금 우리는 바로 그 나라가, 아니 이러한 정신과 신념으로 잉태되고 헌신하는 어느 나라이든지, 과연 오래도록 굳건할 수 있는가 하는 시험대인 거대한 내전에 휩싸여 있습니다.
우리는 바로 그 전쟁의 거대한 싸움터인 이곳에 모여 있습니다.
우리가 여기에 온 것은 바로 그 싸움터의 일부를, 이곳에서 자신의 삶을 바쳐 바로 그 나라를 살리고자 한 영령들의 마지막 안식처로 봉헌하기 위함입니다.
우리의 이 헌정은 지극히 마땅하고 옳습니다.
그러나 더 큰 의미에서 보자면, 우리는 이 땅을 헌정할 수도, 축성할 수도, 신성화할 수도 없습니다.
여기서 싸웠던 용맹한 전사자와 생존 용사들이 이미 이곳을 신성한 땅으로 축성하였기에, 보잘것없는 우리의 힘으로 더 보태고 뺄 것 따위 있을 수 없습니다.
세상은 오늘 우리가 여기 모여 하는 말들을 별로 주목하지도 오래 기억하지도 않을 것이나, 그분들이 이곳에서 이루어낸 것은 결단코 잊을 수 없을 것입니다.
오히려 이 자리에서 우리 살아있는 자들이, 여기서 싸웠던 그분들이 그토록 고결하게 전진시킨 미완의 과업을 수행하는 데 우리 스스로를 봉헌 하여야 합니다.
이 자리에서 우리는 우리 앞에 놓여있는 그 위대한 사명, 즉 고귀한 순국선열들이 마지막 신명을 다 바쳐 헌신했던 그 대의를 위하여 더욱 크게 헌신하여야 하고,
이분들의 죽음을 무위로 돌리지 않으리라 이 자리에서 굳게 결단하여야 하며, 이 나라가 하나님 아래에서 자유의 새로운 탄생을 누려야 할 뿐 아니라, 국민의, 국민에 의한, 국민을 위한 통치가 지상에서 소멸하지 않아야 한다는 그 위대한 사명에 우리 스스로를 바쳐야 합니다.

2. 본론

1) 링컨 대통령 국문 번역 연설문 텍스트 데이터 분석

(2) 토픽분석 결과

단어 빈도표	
우리	12
나라	5
헌신	3
자리	3
국민	3
신념	2
잉태	2
자유	2
정신	2
하게	2
거대	2
싸움터	2
마지막	2
헌정	2
축성	2
들이	2
스스로	2
위대	2

↳ 분석결과 선별된 빈도가 2 이상의 단어 빈도표이다.

2. 본론

1) 링컨 대통령 국문 번역 연설문 텍스트 데이터 분석

(3) 단어 구름 시각화

토픽분석 결과로 최소 2번 이상의 빈도에 해당하는 단어들을 선별하여 빈도가 많은 순서대로 크기를 나타내고, 중앙으로 시각화한 자료는 다음과 같다.

↳ 시각화 결과 링컨의 연설문에서는 '우리', '나라' 등의 단어들이 빈도높게 즉 여러번 강조해서 사용된 것을 볼 수 있다.

2. 본론

1) 링컨 대통령 국문 번역 연설문 텍스트 데이터 분석

(4) 연관어 분석

링컨 대통령 국문 번역 연설문에서 단어를 추출한 뒤 연관규칙에 의해 연관어 분석한 결과는 다음과 같다.

연관어	향상도	연관어	향상도
{헌정}--{우리}	1.1818182	{신념}--{나라}	3.2500000
{싸움터}--{우리}	1.1818182	{나라}--{신념}	3.2500000
{위대}--{우리}	1.1818182	{신념}--{우리}	1.1818182
{스스로}--{우리}	1.1818182	{잉태}--{정신}	6.5000000
{거대}--{우리}	1.1818182	{정신}--{잉태}	6.5000000
{봉헌}--{우리}	1.1818182	{잉태}--{나라}	3.2500000
{자유}--{나라}	3.2500000	{나라}--{잉태}	3.2500000
{나라}--{자유}	3.2500000	{잉태}--{우리}	1.1818182
{헌신}--{우리}	1.1818182	{정신}--{나라}	3.2500000
{신념}--잉태}	6.5000000	{나라}--{정신}	3.2500000
{잉태}--{신념}	6.5000000	{정신}--{우리}	1.1818182
{신념}--{정신}	6.5000000	{자리}--{우리}	0.7878788
{정신}--{신념}	6.5000000	{나라}--{우리}	0.8863636

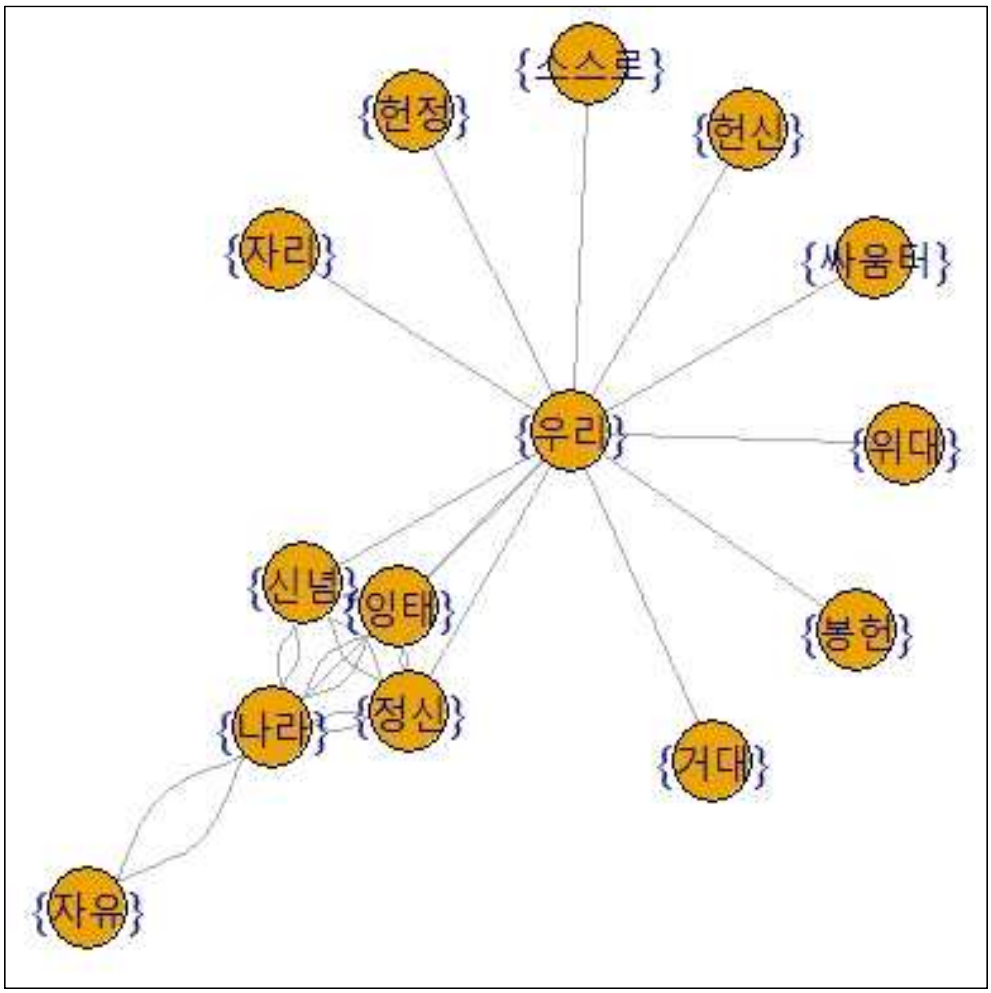
↳ 향상도 기준으로 1보다 높으면 양의 관계이며, 연관성이 매우 높다고 판단한다. 1이면 연관성이 없는 서로 독립적인 관계라고 판단하고, 1보다 작으면 음의 관계이며, 연관성이 없다고 판단한다.

2. 본론

1) 링컨 대통령 국문 번역 연설문 텍스트 데이터 분석

(5) 연관어 시각화

앞서 연관규칙에 의해서 연관어 분석한 결과를 시각화 한 자료는 다음과 같다.



↳ 시각화 결과로 각각의 단어와 단어 사이에 연관이 있는 것들이 선으로 연결되며, '우리'라는 단어가 가장 많은 단어와 연관이 되어있는 것을 볼 수 있다.

2. 본론

2) 다음 포털사이트의 실시간 뉴스 텍스트 데이터 분석

(1) 토픽분석 결과

다음 포털사이트의 실시간 뉴스 토픽분석 결과로 상위 10개의 단어는 다음과 같다.('21. 11. 24, 23시 37분 기준)

사망	5
확진	5
만원	5
김종인	4
전두환	4
산모	3
합류	3
검토	3
보상	3
상향	3

↳ '사망', '확진', '만원' 등의 단어의 빈도수가 5번으로 가장 많은 빈도수를 차지하고 , 그 다음으로는 '김종인', '전두환' 등의 단어들이 있다.

2. 본론

2) 다음 포털사이트의 실시간 뉴스 텍스트 데이터 분석

(2) 단어구름 시각화

앞서 분석한 토픽분석을 시각화한 결과는 다음과 같다.

↳ 가장 많이 나타난 '사망', '확진', '만원' 등의 단어의 크기가 가장 크고 중앙에 표시된다. 이를 통해 가장 이슈인 뉴스가 무엇인지 알아 볼 수 있다.

2. 본론

2) 다음 포털사이트의 실시간 뉴스 텍스트 데이터 분석

(3) 주요이슈 파악

앞서 실시한 토픽 분석 및 단어 구름 시각화를 통해 빈도가 높은 단어들인 '사망', '확진', '김종인', '전두환' 등이 분석 되었다.

이를 통해 주요이슈는 사회·정치 등으로 파악할 수 있다.

3. 부록

사용코드 ; 세부사항은 첨부자료 보고서_코드(주).R 참고

```
install.packages("https://cran.rstudio.com/bin/windows/contrib/3.4/KoNLP_0.80.1.zip",
                 repos = NULL)
install.packages('tm')
install.packages('wordcloud')
install.packages('RColorBrewer')

library(KoNLP)
library(tm)
library(wordcloud)
library(RColorBrewer)



setwd('c:/Rwork/dataset3/dataset3')
lga <- file("lga.txt", encoding = "UTF-8")
lga
lga_data<-readLines(lga)
head(lga_data)

#사용자 정의 함수 작성
exNouns <- function(x) { paste(extractNoun(as.character(x)), collapse = " ") }
# exNouns() 함수를 이용하여 단어 추출
lga_nouns <- sapply(lga_data, exNouns)
lga_nouns[1]


#추출된 단어를 이용하여 말뭉치(Corpus) 생성
myCorpus <-Corpus(VectorSource(lga_nouns))
#데이터 전처리
#문장부호 제거
myCorpusPrepro <- tm_map(myCorpus, removePunctuation)
#수치 제거
myCorpusPrepro <- tm_map(myCorpusPrepro, removeNumbers)
#소문자 변경
myCorpusPrepro <- tm_map(myCorpusPrepro, tolower)
#불용어 제거
myCorpusPrepro <- tm_map(myCorpusPrepro, removeWords, stopwords('english'))
#전처리 결과 확인
inspect(myCorpusPrepro[1:5])
```

```
#전처리된 단어집에서 2 ~ 8 음절 단어 대상 선정
myCorpusPrepro_term <-
  TermDocumentMatrix(myCorpusPrepro,
                     control = list(wordLengths = c(4, 16)))
myCorpusPrepro_term
# matrix 자료구조를 data.frame 자료구조로 변경
myTerm_df <- as.data.frame(as.matrix(myCorpusPrepro_term))
dim(myTerm_df )

### 실습: 단어 출현 빈도수 구하기######
wordResult <- sort(rowSums(myTerm_df), decreasing = TRUE)
wordResult[1:10]
#문장부호 제거
myCorpusPrepro <- tm_map(myCorpus, removePunctuation)
#수치 제거
myCorpusPrepro <- tm_map(myCorpusPrepro, removeNumbers)
#소문자 변경
myCorpusPrepro <- tm_map(myCorpusPrepro, tolower)
#제거할 단어 지정
myStopwords = c(stopwords('english'), "사용", "하기")
#불용어 제거
myCorpusPrepro <- tm_map(myCorpusPrepro, removeWords, myStopwords)
#단어 선별과 평서문 변환
myCorpusPrepro_term <-
  TermDocumentMatrix(myCorpusPrepro,
                     control = list(wordLengths = c(4, 16)))
myTerm_df <- as.data.frame(as.matrix(myCorpusPrepro_term))
#단어 출현 빈도수 구하기
wordResult <- sort(rowSums(myTerm_df), decreasing = TRUE)
wordResult[1:10]
data.frame(wordResult)

##########################################################

#단어 구름에 디자인(빈도수, 색상, 위치, 회전 등) 적용하기
#단어 이름과 빈도수로 data.frame 생성
myName <- names(wordResult)
word.df <- data.frame(word = myName, freq = wordResult)
str(word.df )
# 단어 색상과 글꼴 지정
pal <- brewer.pal(12, "Paired")
#단어 구름 시각화
wordcloud(word.df$word, word.df$freq, scale = c(5, 1),
          min.freq = 2, random.order = F,
          rot.per = .1, colors = pal, family = "malgun")
```



```
##2번
setwd('c:/Rwork/dataset3/dataset3')
lga <- file("lga.txt", encoding = "UTF-8")
lga
lga_data<-readLines(lga)

close(lga)
head(lga_data)
#줄 단위 단어 추출
lword <- Map(extractNoun, lga_data)
length(lword)
lword <- unique(lword)#중복단어 제거
length(lword)
#중복 단어 제거와 추출 단어 확인
lword <- sapply(lword, unique)
length(lword)
lword

filter1 <- function(x) {
  nchar(x) <= 4 && nchar(x) >= 2 && is.hangul(x)
}
filter2 <- function(x) { Filter(filter1, x) }
#줄 단위로 추출된 단어 전처리
lword <- sapply(lword, filter2)
lword

#연관분석을 위한 패키지 설치와 로딩
#install.packages("arules")
library(arules)
# 단계 2: 트랜잭션 생성
wordtran <- as(lword, "transactions")
wordtran

#연관규칙 발견
library(backports)

tranrules <- apriori(wordtran,
                     parameter = list(supp = 0.1, conf = 0.4))
#연관규칙 생성 결과보기
detach(package:tm, unload=TRUE)
inspect(tranrules)

# 연관어 시각화하기
#연관단어 시각화를 위해서 자료구조 변경
rules <- labels(tranrules, ruleSep = " ")
rules
```



```
#문자열로 묶인 연관 단어를 행렬구조로 변경
rules <- sapply(rules, strsplit, " ", USE.NAMES = F)
rules
#행 단위로 묶어서 matrix로 변환
rulemat <- do.call("rbind", rules)
class(rulemat)
#연관어 시각화를 위한 igraph 패키지 설치와 로딩
#install.packages("igraph")
library(igraph)
#edgelist 보기
ruleg <- graph.edgelist(rulemat[c(2:29), ], directed = F)
ruleg

plot.igraph(ruleg, vertex.label = V(ruleg)$name,
            vertex.label.cex = 1.2, vertext.label.color = 'black',
            vertex.size = 20, vertext.color = 'green',
            vertex.frame.co.or = 'blue')

###3번

#install.packages("httr")
library(httr)
#install.packages("XML")
library(XML)

#url요청
url <- "https://news.daum.net"
web <- GET(url)
web

#HTML 파싱하기
html <- htmlTreeParse(web, useInternalNodes = T, trim = T, encoding = "utf-8")
rootNode <- xmlRoot(html)

#태그 자료 수집하기
news <- xpathSApply(rootNode, "//a[@class = 'link_txt']", xmlValue)
news

#자료 전처리 - 수집한 문서를 대상으로 불용어 제거
news_pre <- gsub("[\r\n\t]", ' ', news)
news_pre <- gsub('[[:punct:]]', ' ', news_pre)
news_pre <- gsub('[[:cntrl:]]', ' ', news_pre)
news_pre <- gsub('\\d+', ' ', news_pre) # corona19(covid19) 때문에 숫자 제거 생략
news_pre <- gsub('[a-z]+', ' ', news_pre)
news_pre <- gsub('[A-Z]+', ' ', news_pre)
news_pre <- gsub('\\s+', ' ', news_pre)
news_pre
```



```
#기사와 관계 없는 'TODAY', '검색어 순위' 등의 내용은 제거
news_data <- news_pre[1:62]
news_data

#수집한 자료를 파일로 저장하고 읽기
setwd("C:/Rwork/data/")
write.csv(news_data, "news_data1.csv", quote = F)
news_data <- read.csv("news_data.csv", header = T, stringsAsFactors = F)
str(news_data)
names(news_data) <- c("no", "news_text")
head(news_data)
news_text <- news_data$news_text
news_text

#사용자 정의 함수 작성
exNouns <- function(x) { paste(extractNoun(x), collapse = " ")}
# exNouns() 함수를 이용하여 단어 추출
news_nouns <- sapply(news_text, exNouns)
news_nouns
#추출 결과 확인
str(news_nouns)


library(tm)

#추출된 단어를 이용한 말뭉치(corpus) 생성
newsCorpus <- Corpus(VectorSource(news_nouns))
newsCorpus
inspect(newsCorpus[1:5])
#단어 vs 문서 집계 행렬 만들기##
TDM <- TermDocumentMatrix(newsCorpus, control = list(wordLengths = c(4, 16)))
TDM
#matrix 자료구조를 data.frame 자료구조로 변경
tdm.df <- as.data.frame(as.matrix(TDM))
dim(tdm.df )
#단어 출현 빈도수 구하기
wordResult <- sort(rowSums(tdm.df), decreasing = TRUE)
wordResult[1:10]
##################
library(wordcloud)
myNames <- names(wordResult)
myNames
# 단어와 단어 빈도수 구하기
df <- data.frame(word = myNames, freq = wordResult)
head(df )
#단어 구름 생성
pal <- brewer.pal(12, "Paired")
wordcloud(df$word, df$freq, min.freq = 2,
          random.order = F, scale = c(4, 0.7),
          rot.per = .1, colors = pal, family = "malgun")
```



```
#install_github("lchiffon/wordcloud2")
library(wordcloud2)


wc2data <- data.frame(df$word, df$freq)
wc2data

wordcloud2(data=df, size=0.5, color='random-light', backgroundColor = "black")
```



4. 참고자료

- ○ 보고서_코드(주).R
- ○ 보고서_코드(보조1).R
- ○ 보고서_코드(보조2).R
- ○ Lincoln.txt
- ○ 뉴스 자료 사이트 ; "https://news.daum.net"