# 연설문 Topic Modeling

송경렬

## Libraries

```
library(readr)
library(dplyr)
library(stringr)
library(textclean)
library(tidytext)
library(KoNLP)

library(topicmodels)

library(ggplot2)
library(ldatuning)
library(scales)
library(data.table)
```

## 데이터 로드

- read.csv()로 읽었을 때 한글이 자꾸 깨지는 문제가 발생하여 그 대안으로 fread()를 사용하였습니다.

```
setwd("C:/Users/thd94/OneDrive/바탕 화면/textminig/Data/Data")

raw_speeches<-fread("speeches_roh3.csv",encoding="UTF-8")

raw_speeches<-raw_speeches[c(order(raw_speeches$id)),]
```

## 전처리

- 한글이 아닌 문자를 제거하고 연속해서 나오는 공백을 제거합니다.
- 증복되는 연설 문장들을 모두 제거해줍니다.
- 단어가 3 개 이상 포함되어 있는 연설 문장만 뽑아줍니다.

```
new_roh<-raw_speeches %>%
  mutate(content = str_replace_all(string=content,pattern="[^가-힣]",replacement=" "),
         content = str_squish(content)) %>%
  distinct(content,.keep_all=T) %>%
  filter(str_count(content,boundary("word"))>=3)
```

## 명사 추출

- 앞선 처리 이후 명사만을 추출합니다.
- 명사를 추출하고 빈도가 100 회 미만인 단어만 추출합니다.

```
speech_content<-new_roh %>%
  unnest_tokens(input = content,
                output = word,
                token = extractNoun,
                drop = F) %>%
  filter(str_count(word)>1) %>%
  group_by(id) %>%
  distinct(word,.keep_all=T) %>%
  ungroup() %>%
  select(id,word)

# 빈도가 100 회 이하인 단어만 추출
count_word<-speech_content %>%
  add_count(word) %>%
  filter(n<=100) %>%
  select(-n)
```

## 불용어

- 불용어를 고르기 위해 빈도가 높은 단어 상위 400 개를 살펴봅니다.
- 불용어들을 선택합니다.
- 불용어를 제거해주고, 유의어들을 처리해줍니다.

```
count_word %>%
  count(word,sort=T) %>%
  print(n=400)



stopword<-c("다음","당시","경우","되기","불과","하시","얼마","년대","곳곳","로운
","얼마"
            ,"하자","들이","하다","하게","해서","이번","하네","해요","이것","하기
","한거"
            ,"그것","여기","까지","하신","민큼")
```

```r
# 불용어 제거하기, 유의어 처리하기
count_word<-count_word %>%
  filter(!word %in% stopword) %>%
  mutate(word=recode(word,
                     "한국이"="한국",
                     "한국을"="한국",
                     "한반도에"="한반도",
                     "한국에"="한국",
                     "한국의"="한국",
                     "사람들"="사람",
                     "양국간"="양국"))
```

# 최적의 Topic 수 찾기

## LDA 모델 만들기

- 문서별 단어 빈도를 구하고, DTM 을 만들어 줍니다.
- 최적의 Topic 수를 찾고, 그래프로 확인합니다.

```r
# 문서별 단어 빈도 구하기
count_word_doc<-count_word %>%
  count(id,word,sort=T)

count_word_doc
```

```
## # A tibble: 76,299 x 3
##       id word       n
##    <int> <chr>  <int>
##  1    32 한국       3
##  2    63 한국       3
##  3    69 한국       3
##  4   104 한국       3
##  5   313 한국       3
##  6   362 한국       3
##  7   443 한국       3
```

```
##  8   469 한국       3
##  9   600 한국       3
## 10     2 한반도     2
## # ... with 76,289 more rows
```

```
# DTM
dtm_speeches<-count_word_doc %>%
  cast_dtm(document=id,term=word,value=n)
dtm_speeches
```

```
## <<DocumentTermMatrix (documents: 757, terms: 13345)>>
## Non-/sparse entries: 76299/10025866
## Sparsity           : 99%
## Maximal term length: 13
## Weighting          : term frequency (tf)
```

```
as.matrix(dtm_speeches[1:8,1:8])
```

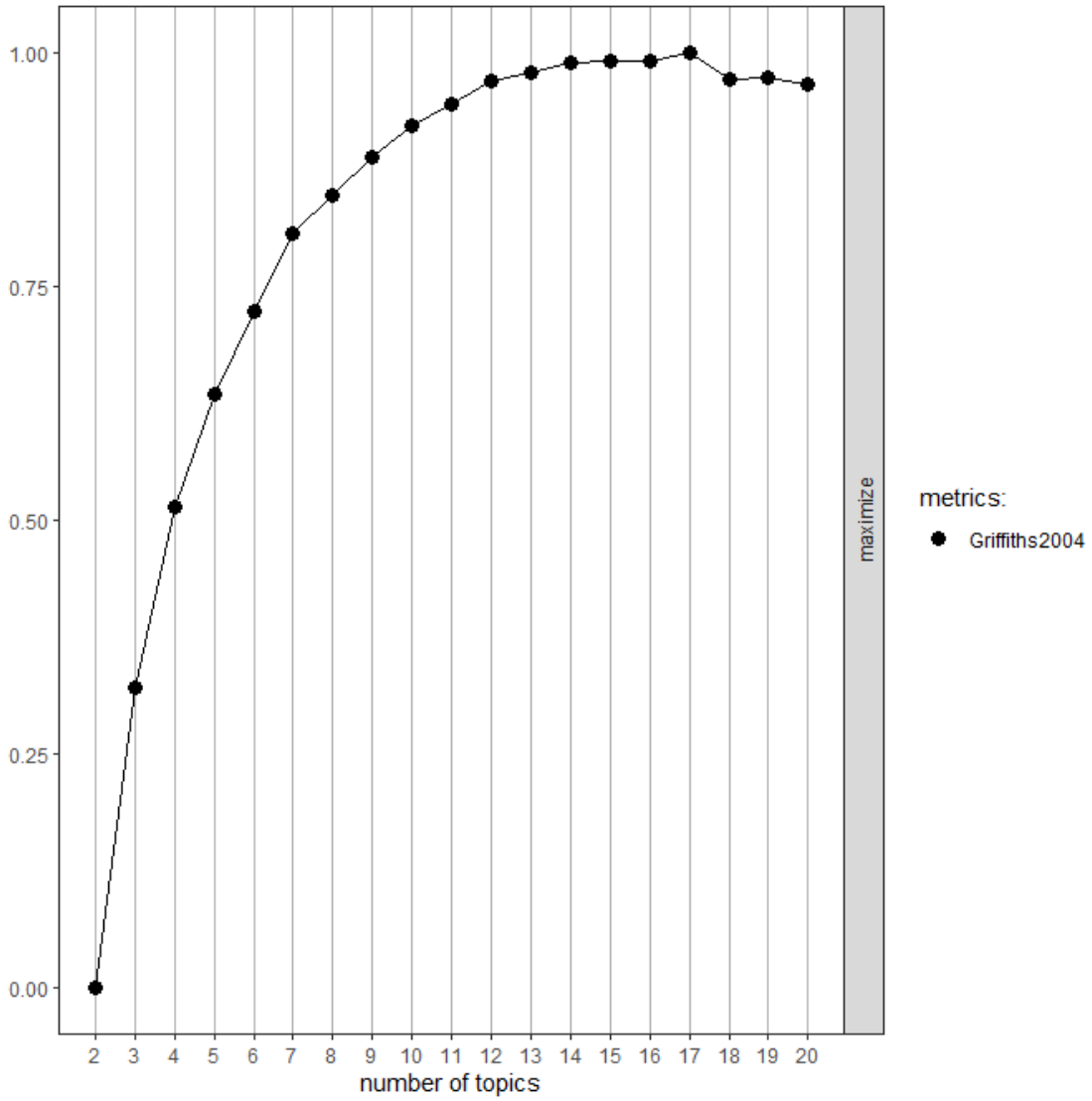
```
##      Terms
## Docs  한국 한반도 갈림길 감당 강력 건배를 경제시스템 공조
##   32     3      2      1    0    0      0          0    0
##   63     3      2      0    0    0      0          1    0
##   69     3      0      0    0    1      0          0    0
##   104    3      1      0    0    0      0          1    0
##   313    3      0      0    0    0      0          0    0
##   362    3      0      0    0    0      0          0    0
##   443    3      0      0    0    0      0          0    0
##   469    3      0      0    0    0      0          0    1
```

```
# 최적의 토픽 수 찾기

models<-FindTopicsNumber(dtm=dtm_speeches,
                         topics=2:20,
                         return_models=T,
                         control=list(seed=1234))
```

## 최적의 Topic 수 그래프

```
FindTopicsNumber_plot(models)
```




## 모델 추출

```
optional_model<-models %>%
  filter(topics==17) %>%
  pull(LDA_model) %>%
  .[[1]]
```

# 각 Topic 마다 등장확률이 높은 상위 10 개 단어추출 & 토픽별 주요단어

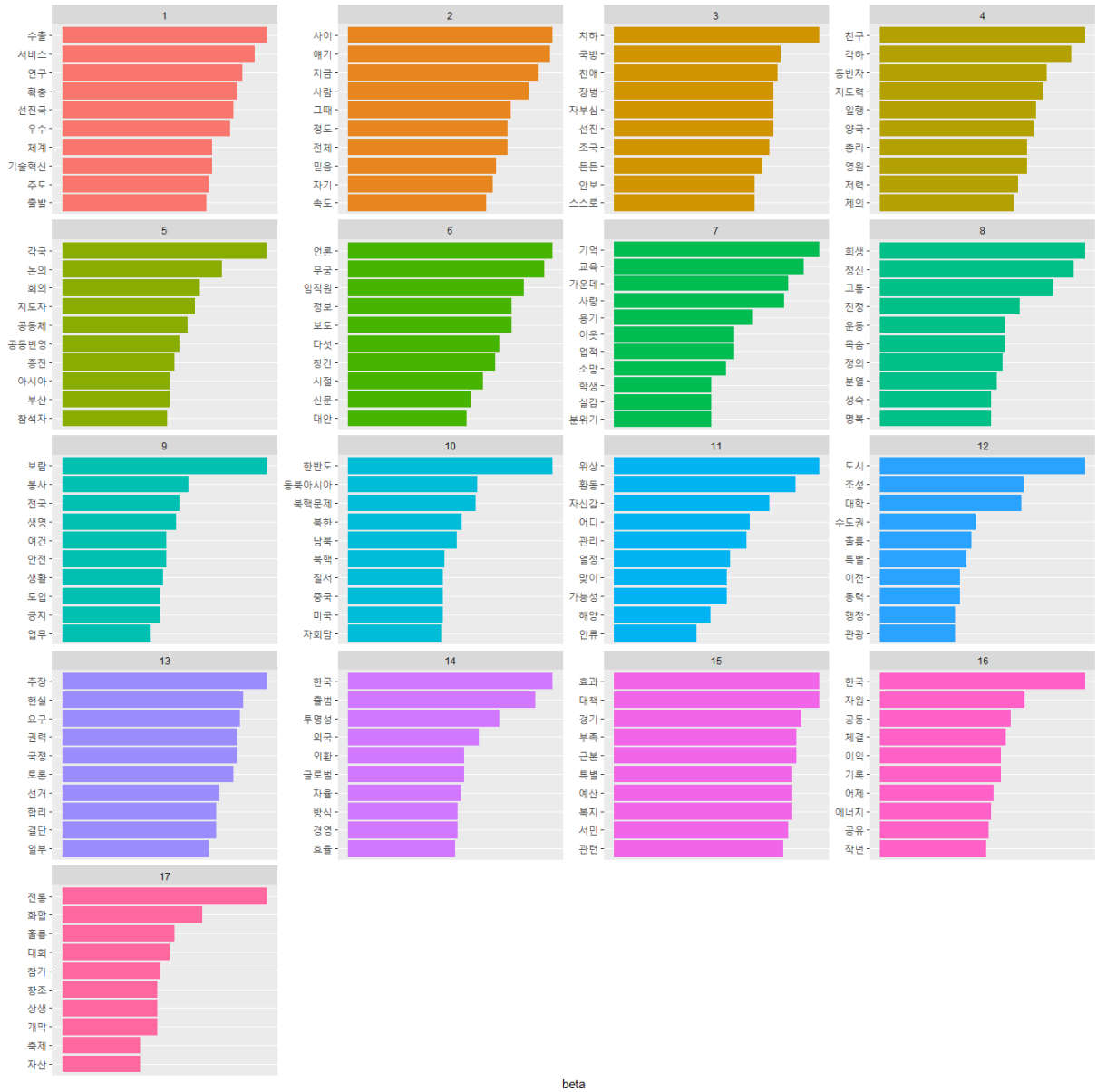
## Topic 별로 beta 가 가장 높은 단어 추출

```
term_topic<-tidy(optional_model,matrix="beta")

top_term_topic<-term_topic %>%
  group_by(topic) %>%
  slice_max(beta,n=10)
```

## 그래프 그리기

```
ggplot(top_term_topic,aes(x=reorder_within(term,beta,topic),y=beta,fill=factor(topic)))+
  geom_col(show.legend=F)+
  facet_wrap(~topic,scales="free",ncol=4)+
  coord_flip()+
  scale_x_reordered()+
  scale_y_continuous(breaks=NULL)+
  labs(x=NULL)
```



# 원문을 토픽으로 분류한 후 토픽별 문서 수와 주요 단어 나타내기

## 문서별로 gamma 가 가장 높은 토픽 추출

```
doc_topic<-tidy(optional_model,matrix="gamma")
doc_class<-doc_topic %>%
  group_by(document) %>%
  slice_max(gamma,n=1)

doc_class %>% head(10)
```

```
## # A tibble: 10 x 3
## # Groups:   document [10]
##    document topic  gamma
##    <chr>    <int>  <dbl>
##  1 1           10 0.213
##  2 10           8 0.222
##  3 100          3 0.348
##  4 101          8 0.0951
##  5 102          2 0.196
##  6 103          5 0.375
##  7 104          5 0.254
##  8 105         15 0.420
##  9 106          5 0.335
## 10 107         17 0.199
```

## 원문에 가장 높은 토픽 번호 부여하기

```
doc_class$document<-as.integer(doc_class$document)

speeches_topic<-raw_speeches %>%
  left_join(doc_class,by=c("id"="document"))

speeches_topic<-speeches_topic %>% na.omit()
```

## 토픽별 문서 수와 단어 시각화

- 토픽별 주요 단어 목록을 만듭니다.
- 토픽별 문서 빈도를 구합니다.
- 문서 빈도에 주요 단어를 결합합니다.
- 토픽별 문서 수와 주요 단어로 막대 그래프를 만듭니다.

### 1. 토픽별 주요 단어 목록

```
top_terms<-term_topic %>%
  group_by(topic) %>%
  slice_max(beta,n=6,with_ties=F) %>%
  summarise(term=paste(term,collapse=","))

top_terms %>% head(8)
```

```
## # A tibble: 8 x 2
##   topic term
##   <int> <chr>
## 1     1 수출,서비스,연구,확충,선진국,우수
## 2     2 사이,얘기,지금,사람,그때,정도
## 3     3 치하,국방,친애,선진,장병,자부심
## 4     4 친구,각하,동반자,지도력,일행,양국
## 5     5 각국,논의,회의,지도자,공동체,공동번영
## 6     6 언론,무궁,임직원,정보,보도,다섯
## 7     7 기억,교육,가운데,사랑,용기,업적
## 8     8 희생,정신,고통,진정,운동,목숨
```

### 2. 토픽별 문서 빈도

```
count_topic <-speeches_topic %>%
  count(topic)

count_topic %>% head(8)
```

```
##    topic   n
## 1:     1  47
## 2:     2  30
## 3:     3  57
## 4:     4 108
## 5:     5  44
```

```
## 6:     6  43
## 7:     7  24
## 8:     8  73
```

### 3. 문서 빈도에 주요 단어 결합

```
count_topic_word<-count_topic %>%
  left_join(top_terms,by = "topic") %>%
  mutate(topic_name = paste("Topic",topic))

count_topic_word %>% head(8)
```

```
##    topic   n                                    term topic_name
## 1:     1  47     수출,서비스,연구,확충,선진국,우수    Topic 1
## 2:     2  30         사이,얘기,지금,사람,그때,정도    Topic 2
## 3:     3  57       치하,국방,친애,선진,장병,자부심    Topic 3
## 4:     4 108     친구,각하,동반자,지도력,일행,양국    Topic 4
## 5:     5  44 각국,논의,회의,지도자,공동체,공동번영    Topic 5
## 6:     6  43       언론,무궁,임직원,정보,보도,다섯    Topic 6
## 7:     7  24       기억,교육,가운데,사랑,용기,업적    Topic 7
## 8:     8  73         희생,정신,고통,진정,운동,목숨    Topic 8
```

## 4. 막대그래프

```
ggplot(count_topic_word,
       aes(x=reorder(topic_name,n),y=n,fill=topic_name))+
  geom_col(show.legend=F)+
  coord_flip()+
  geom_text(aes(label=n),hjust=-0.2)+
  geom_text(aes(label=term),hjust=1.03,col="white",fontface="bold")+
  scale_y_continuous(expand=c(0,0),limits=c(0,130))+
  labs(x=NULL)
```