

부록 : 각각의 분석 단계에 대한 R text mining source code.

package loading

```
if(!require(pacman)) {install.packages("pacman")}

pacman::p_load(dplyr, data.table, tidytext, tidyr,
               ggplot2, wordcloud, wordcloud2, tm,
               openNLP, bibliometrix, igraph, ggraph)

pacman::p_load_gh('hrbrmstr/pluralize')
path_csv <- paste0(dir("csv"))
```

원시자료 전처리

```
data_list_csv <- lapply(paste0("csv/",path_csv), fread)
data_list_csv[[5]][,ISBN:=as.character(ISBN)]
data_list_csv[[6]][,ISBN:=as.character(ISBN)]
data_list_csv[[7]][,ISBN:=as.character(ISBN)]
raw_csv <- rbindlist(data_list_csv)
raw_csv$Abstract[raw_csv$Abstract=="[No abstract available]"] <- NA
rm(data_list_csv)
```

자료 구조 확인

```
str(raw_csv)
```

Unigram Token화

```
tut <- raw_csv %>%
  unnest_tokens(word, Title) %>%
  filter(!word %in% stop_words$word)
```

단수화(Singularizing) 전의 단어 빈도

```
tut %>%
  count(word, sort = T)
```

```
## # A tibble: 8,176 x 2
##       word      n
##   <chr> <int>
## 1   wave  3542
## 2  waves  1771
## 3   model  1132
## 4 numerical 1060
## 5  analysis 1012
## 6   study   887
## 7   water   881
## 8   method  621
## 9    flow   604
## 10 experimental 564
## # ... with 8,166 more rows
```

단수화(Singularizing)

```
tut$word <- singularize(tut$word)
```

품사부여(POS tagging, Part of Speech tagging) 및 추출

```
unigram_title <- tut %>%
  count(word, sort = T) %>%
  merge(., parts_of_speech, by = "word") %>%
  arrange(desc(n)) %>%
  filter(pos == "Noun" | pos == "Adjective", n > 100)
```

Bigram Token화 및 가공

```
tbt_count <- raw_csv %>% select(Title) %>%
  unnest_tokens(bigram, Title, token = "ngrams", n = 2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  count(word1, word2, sort = T)

tbt_count$word1 <- singularize(tbt_count$word1)
tbt_count$word2 <- singularize(tbt_count$word2)

tbt <- tbt_count %>%
  unite(bigram, word1, word2, sep = " ")
```

Unigram word cloud(Fig. 1)

```
unigram_title %>%
  filter(!word %in% c("study", "analysis", "water",
                     "effect", "method", "sea")) %>%
  with(wordcloud(word, n, max.words = 150, random.order = F,
                 scale = c(5,.8),
                 colors = brewer.pal(8, "Dark2")))
```

빈도 600 이상의 Unigram 시각화(Fig. 2)

```
unigram_title %>%
  filter(n > 600,
         !word %in% c("study", "analysis", "water",
                     "effect", "method", "sea")) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col() +
  ggtitle("Word frequency in Coastal & Ocean Engineering articles") +
  xlab("words") +
  ylab("frequency") +
  coord_flip() +
  theme(title = element_text(size = 20, face = "bold"),
        axis.text=element_text(size=12),
        axis.title=element_text(size=16,face="bold"))
```

Unigram word cloud(Fig. 3)

```
unigram_title %>%
  filter(!word %in% c("wave", "study", "analysis",
                     "water", "effect", "method", "sea")) %>%
  with(wordcloud(word, n, max.words = 150, random.order = F,
                 scale = c(5,.8),
                 colors = brewer.pal(8, "Dark2")))
```

Bigram word cloud(Fig. 4)

```
tbt %>%
  with(wordcloud(bigram, n, max.words = 50, random.order = F,
                 scale = c(3,.4),
                 colors = brewer.pal(8, "Dark2")))
```

빈도 100 이상의 Bigram 시각화(Fig. 5)

```
tbtf %>%
  filter(n > 100,
    !bigram %in% c("water wave", "numerical simulation",
      "experimental study", "numerical study",
      "numerical model", "experimental investigation",
      "finite element", "shallow water")) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n)) +
  geom_col() +
  ggtitle("Word frequency in Coastal & Ocean Engineering articles") +
  xlab("bigram") +
  ylab("frequency") +
  coord_flip() +
  theme(title = element_text(size = 20, face = "bold"),
    axis.text = element_text(size = 15),
    axis.title = element_text(size = 18, face = "bold"))
```

Bigram word network(Fig. 6)

```
bigram_graph <- tbtf_count %>%
  filter(n > 50) %>%
  graph_from_data_frame()

a <- grid::arrow(type = "closed", length = unit(.15, "inches"))

ggraph(bigram_graph, layout = "auto") +
  geom_edge_link(aes(edge_alpha = n*100), show.legend = F,
    arrow = a, end_cap = circle(.07, "inches")) +
  geom_node_point(color = "lightblue", size = 5) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 0.5) +
  theme(axis.line = element_blank(),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank(),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    legend.position = "none",
    #panel.background = element_blank(),
    panel.background = element_rect(colour = "lightgray"),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    plot.background = element_blank())
```