인문학 텍스트 마이닝

Text Analysis utilizing twitter

Text Analysis utilizing twitter

- 라이브러리 설치

```
> library(bitops)
> library(RCurl)
> library(RJSONIO)
> library(twitteR)
> library(ROAuth)
> library(RColorBrewer)
> library(devtools)
> install_github("twitteR", username="geoffjentry")
Downloading github repo geoffjentry/twitteR@master
Installing twitteR
'/Library/Frameworks/R.framework/Resources/bin/R' --vanilla CMD INSTALL  \

'/private/var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T/Rtmp8qGMiY/devtoolscb924cc3a7ae/geoffj
entry-twitteR-563a23c'  \
  --library='/Library/Frameworks/R.framework/Versions/3.1/Resources/library'  \
  --install-tests

* installing *source* package ‘twitteR’ ...
** R
** inst
** preparing package for lazy loading
Creating a generic function for ‘as.data.frame’ from package ‘base’ in package ‘twitteR’
** help
*** installing help indices
** building package indices
** testing if installed package can be loaded
* DONE (twitteR)
Reloading installed twitteR

Attaching package: ‘twitteR’

The following object is masked from ‘package:plyr’:

    id

The following objects are masked from ‘package:dplyr’:

    id, location

경고메시지:
Username parameter is deprecated. Please use geoffjentry/twitteR
```

GitHub에서 twitteR패키지의 최신버젼을 다운로드한다.

Text Analysis utilizing twitter

- 유저 정보 입력

```
> api_key <- "[redacted]"
>
> api_secret <- "[redacted]"
>
> access_token <- "[redacted]"
>
> access_token_secret <- "[redacted]"
>
> setup_twitter_oauth(api_key,api_secret,access_token,access_token_secret)
[1] "Using direct authentication"
```

- https://apps.twitter.com에서 로그인 후 제공받은 api_key, api_secret, access_token, access_token_secret을 입력한다.

Text Analysis utilizing twitter

- 긍부정 분류함수

```
> score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
+ {
+     require(plyr)
+     require(stringr)
+
+     # we got a vector of sentences. plyr will handle a list or a vector as an "l" for us
+     # we want a simple array of scores back, so we use "l" + "a" + "ply" = laply:
+     scores = laply(sentences, function(sentence, pos.words, neg.words) {
+
+         # clean up sentences with R's regex-driven global substitute, gsub():
+         sentence = gsub('[[:punct:]]', '', sentence)
+         sentence = gsub('[[:cntrl:]]', '', sentence)
+         sentence = gsub('\\d+', '', sentence)
+         # and convert to lower case:
+         sentence = tolower(sentence)
+
+         # split into words. str_split is in the stringr package
+         word.list = str_split(sentence, '\\s+')
+         # sometimes a list() is one level of hierarchy too much
+         words = unlist(word.list)
+
+         # compare our words to the dictionaries of positive & negative terms
+         pos.matches = match(words, pos.words)
+         neg.matches = match(words, neg.words)
+
+         # match() returns the position of the matched term or NA
+         # we just want a TRUE/FALSE:
+         pos.matches = !is.na(pos.matches)
+         neg.matches = !is.na(neg.matches)
+
+         # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
+         score = sum(pos.matches) - sum(neg.matches)
+
+         return(score)
+     }, pos.words, neg.words, .progress=.progress )
+
+     scores.df = data.frame(score=scores, text=sentences)
+     return(scores.df)
+ }
```

Score.sentiment함수를 입력하여 준다.

Text Analysis utilizing twitter

- Greece에 관련된 텍스트 1000개 크롤링

```
> Greece.tweets = searchTwitter("Greece" , n = 1000)
```

- Greece에 관련된 텍스트만 추출

```
> library(plyr)
>
> Greece.text = laply(Greece.tweets,function(t)t$getText())
```

- 긍부정 단어가 들어있는 사전 불러오기

```
> getwd()
[1] "/Users/Seongmin_M/Downloads"
>
> setwd("/Users/Seongmin_M/Downloads")
>
> pos.words= scan("positive-words.txt",what="character",comment.char=";")
Read 2006 items
>
> neg.words = scan("negative-words.txt",what="character",comment.char=";")
Read 4783 items
```

Text Analysis utilizing twitter

- 긍부정 사전에 단어 추가

```
> pos.words <- c(pos.words,'upgrade')
>
> neg.words <- c(neg.words,'wait','waiting')
```

- 텍스트가 깨지지 않게 문자 인코딩 방식을 UTF-8로 변환

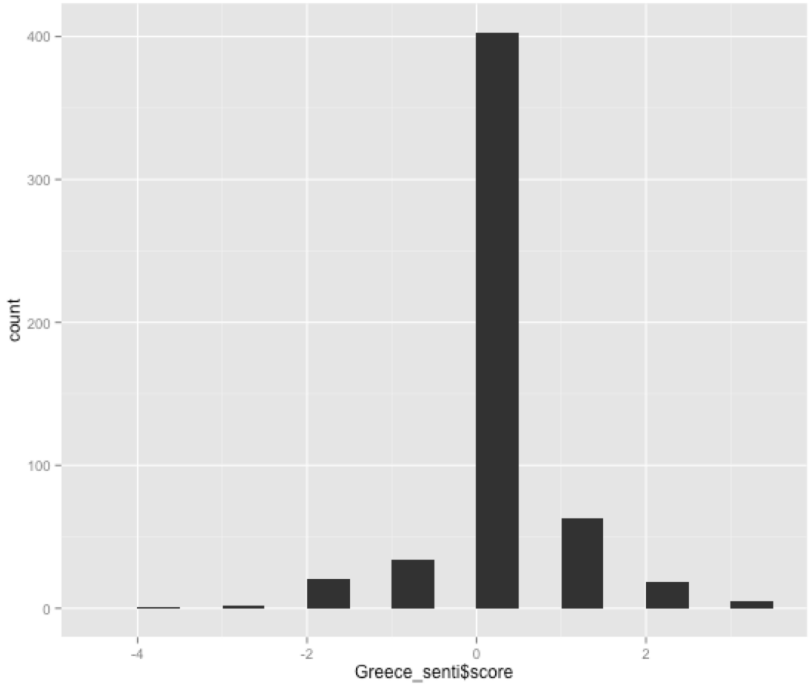
```
> Greece.text = Greece.text[!Encoding(Greece.text)=="UTF-8"]
```

- Greece에 관련 텍스트를 긍부정 단어 사전을 사용하여 분류하기

```
> Greece_senti =score.sentiment(Greece.text,pos.words,neg.words,.progress='text')
  |=================================================================================| 100%
```

- 히스토그램 생성

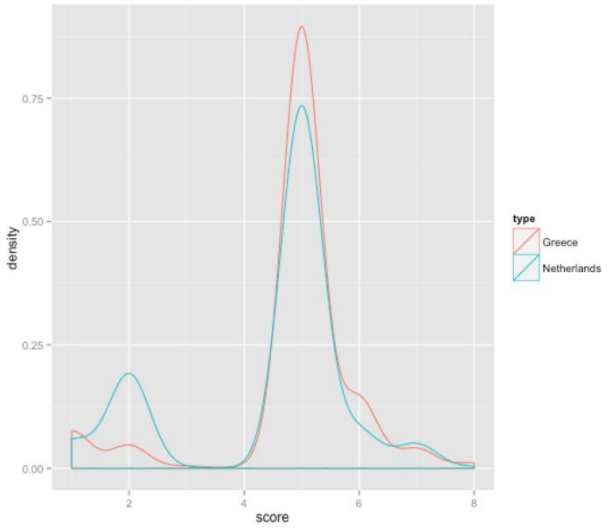
```
> library(ggplot2)
> qplot(Greece_senti$score,binwidth=0.5)
```



Text Analysis utilizing twitter

- Greece와 Netherlands간의 긍부정 비교

```
> a<-dim(Greece_senti)[1]
>
> b<-dim(Netherlands_senti)[1]
>
> country<-rbind(as.data.frame(cbind(type=rep("Greece",a),score=Greece_senti[,1])),as.data.frame
(cbind(type=rep("Netherlands",b),score=Netherlands_senti[,1])))
>
> country$type<-factor(country$type)
>
> country$score<-as.integer(country$score)
>
> ggplot(country,aes(x=score,colour=type))+geom_density()
```



트위터 텍스트를 활용하여 두 나라간 긍부정 반응을 비교한 결과 그리스에 비해 네덜란드에 대해 더 긍정적 반응을 보이는 것을 확인 하였다.

WordCloud utilizing twitter

WordCloud utilizing twitter

- 워드 클라우드 생성

- 모든 문자 소문자로 변환

```
> Greece.text <- tolower(Greece.text)
>
```

- Rt를 빈공간으로 바꾸기(삭제)

```
> Greece.text <- gsub("rt", "", Greece.text)
>
```

- 유저이름 삭제(@||w+)

```
> Greece.text <- gsub("@\\w+", "", Greece.text)
>
```

- 문장 부호 제거

```
> Greece.text <- gsub("[[:punct:]]", "", Greece.text)
>
```

- 링크 제거

```
> Greece.text <- gsub("http\\w+", "", Greece.text)
>
```

WordCloud utilizing twitter

- 워드 클라우드 생성

- 탭 제거

```
> Greece.text <- gsub("[ |\t]{2,}", "", Greece.text)
>
```

- 시작 부분의 문자 제거

```
> Greece.text <- gsub("^ ", "", Greece.text)
>
```

- 끝 부분의 문자 제거

```
> Greece.text <- gsub(" $", "", Greece.text)
```

- TM라이브러리 설치

```
> install.packages("tm")
URL 'http://cran.rstudio.com/bin/macosx/contrib/3.1/tm_0.6.tgz'을 시도합니다
Content type 'application/x-gzip' length 647048 bytes (631 Kb)
URL을 열었습니다
==================================================
downloaded 631 Kb


The downloaded binary packages are in
        /var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpTkkSha/downloaded_packages
>
> library("tm")
필요한 패키지를 로딩중입니다: NLP
```

WordCloud utilizing twitter

- 워드 클라우드 생성

  - Corpus생성

```
>
> Greece.text.corpus <- Corpus(VectorSource(Greece.text))
```

  - Tm_map을 활용하여 Stop words 삭제

```
> Greece.text.corpus <- tm_map(Greece.text.corpus, function(x)removeWords(x,stopwords()))
```

  - 워드클라우드 생성

```
> install.packages("wordcloud")
URL 'http://cran.rstudio.com/bin/macosx/contrib/3.1/wordcloud_2.5.tgz'을 시도합니다
Content type 'application/x-gzip' length 117239 bytes (114 Kb)
URL을 열었습니다
==================================================
downloaded 114 Kb


The downloaded binary packages are in
        /var/folders/28/g8cf_pvx46s5phqgwr6qq7jw0000gn/T//RtmpTkkSha/downloaded_packages
>
> library(wordcloud)
필요한 패키지를 로딩중입니다: RColorBrewer
>
> wordcloud(dalta.text.corpus,min.freq = 2, scale=c(7,0.5),colors=brewer.pal(8, "Dark2"),  rando
m.color= TRUE, random.order = FALSE, max.words = 150)
```

Relation Analysis using Polarity Words

- Clustering Analysis & MDS Visualization

Relation Analysis using Polarity Words

- 긍부정 값을 이용한 관계 분석

- 데이터 가져오기

```
> setwd("/Users/Seongmin_M/Desktop/Class")
>
> data.1=read.csv("ECC_total.csv",header=T)
```

```
> data.1
       Country Chair_count Population    Area X.4 X.3 X.2 X.1  X0  X1  X2 X3 X4 X5 Text_count
1       Greece          72   11125179 131990   1   1   6  40 345  35  23  5  3  0        459
2  Netherlands          25   16372715  41526   0   1   5  29 740  53  10  0  0  0        838
3      Denmark          13    5457415  43094   0   6  12  63 399  80  15  1  1  0        577
4      Germany          99   82314906 357050   0   1  27  48 282 166  23  0  1  0        548
5       Latvia           8    2281305  64589   0   2   7  84 236  44  11  6  1  0        391
6      Romania          33   22276056 238391   0   7   6  42 343  85   9  1  0  0        493
7    Luxemburg           6     476200   2586   0   1   2  37 404  17   2  0  1  0        464
8    Lithuania          12    3373991  65303   0   1  10  55 256  75  16  2  0  0        415
9        Malta           6     404962    316   0   0   5  25 278 187  10  0  2  0        507
10     Belgium          22   10392226  30528   1   0  10  53 344 114  28  8  0  0        558
11    Bulgaria          17    7322858 110910   4   1   4  51 359  84  23  4  1  0        531
12      Sweden          18    9142817 449964   0   3  19  52 315  87  17  3  0  0        496
13       Spain          50   45116894 506030   0   0   6  26 280  69  13  3  0  0        397
14    Slovakia          13    5396168  49037   0   1  37  51 370  78  12  2  2  0        553
15    Slovenia           7    2013597  20273   0   0   1  31 465 129  13  6  1  0        646
16     Ireland          12    4239848  70273   0   1   9  82 347 162  39  6  2  0        648
17     Estonia           6    1342409  45226   0   0   7  47 323 109  13  3  1  0        503
18     Britain          72   60587300 244820   0   5  27  89 229 130  59  4  0  0        543
19     Austria          17    8199783  83871   6   2  36  47 299  82  10  1  0  0        483
20      Italia          72   59131287 301318   0   0   0  10 315  19   1  0  0  0        345
21       Czech          22   10306709  78866   0   1   4 106 344  91  11  1  0  0        558
22     Croatia          12    4398150  56594   0   2   1  26 263 217 202 16  0  0        727
23      Kypros           6     766400   9251   0   0   0   1  40   1   0  0  0  0         42
24    Portugal          22   10599095  92391   0   0   2  12 234 190   7  2  2  0        449
25      Poland          50   38116486 312683   0   0   7  40 431 109  17  3  2  0        609
26      France          72   63392140 674843   0   0   6  42 236  30  24  1  0  0        339
27     Finland          13    5289128 338145   0   1  14  54 380  78  21  2  0  0        550
28     Hungary          22   10066158  93030   0   2  12  55 505  66  10  1  0  0        651
```

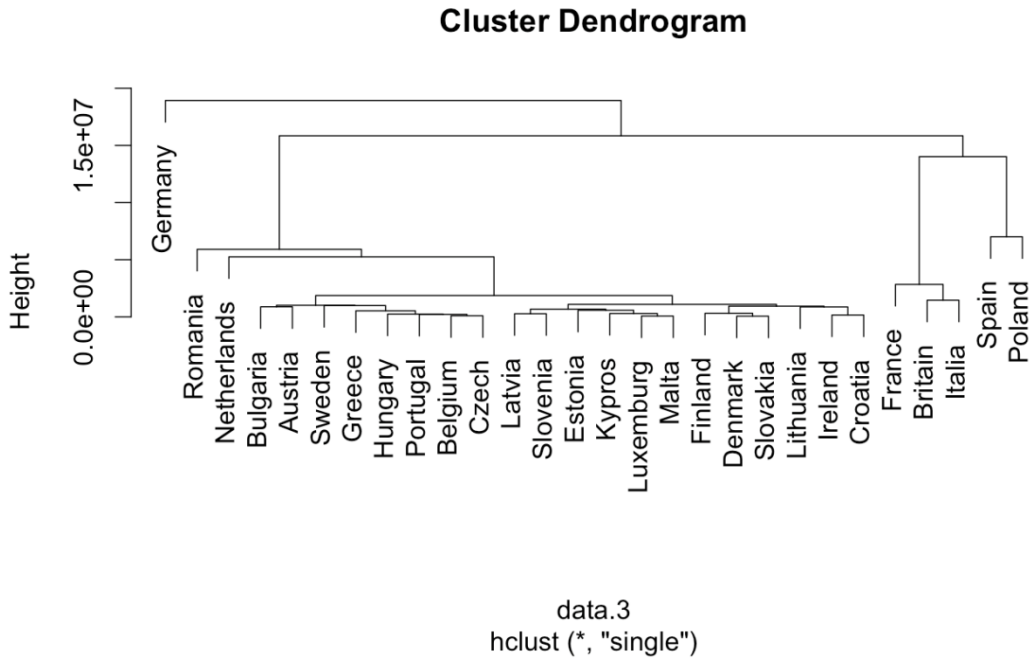
Relation Analysis using Polarity Words

- 행 이름 설정, 데이터 추출, 거리행렬 생성

```
> row.names(data.1)=data.1[,1]
>
> data.2=data.1[,2:15]
>
> data.3=dist(data.2,method="euclidean")
```

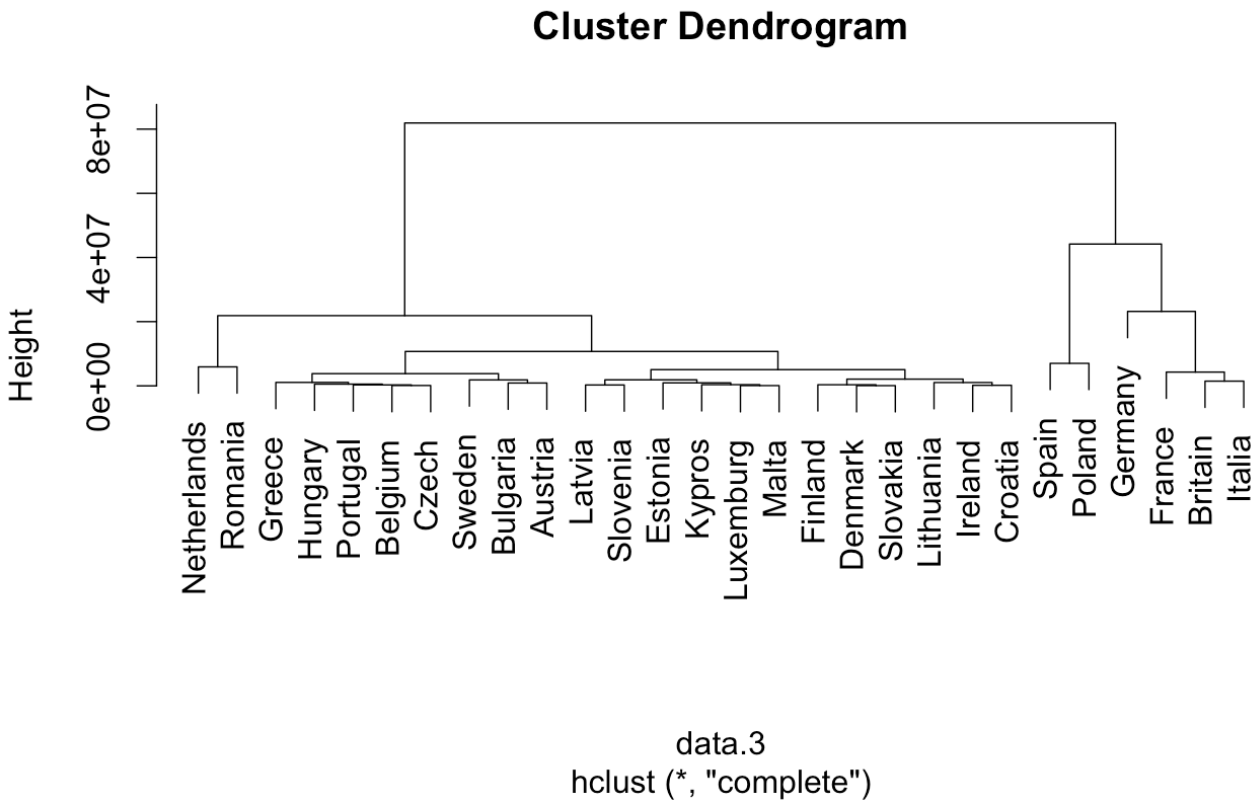
- 유클리디안 거리, 최단 연결법

```
> data.3_1=hclust(data.3,method="single")
>
> png("total_dendrogram_short")
>
> plot(data.3_1)
>
> dev.off()
null device
          1
```



- 유클리디안 거리, 최장 연결법

```
> data.3_2=hclust(data.3,method="complete")
>
> png("total_dendrogram long")
>
> plot(data.3_2)
>
> dev.off()
null device
          1
```



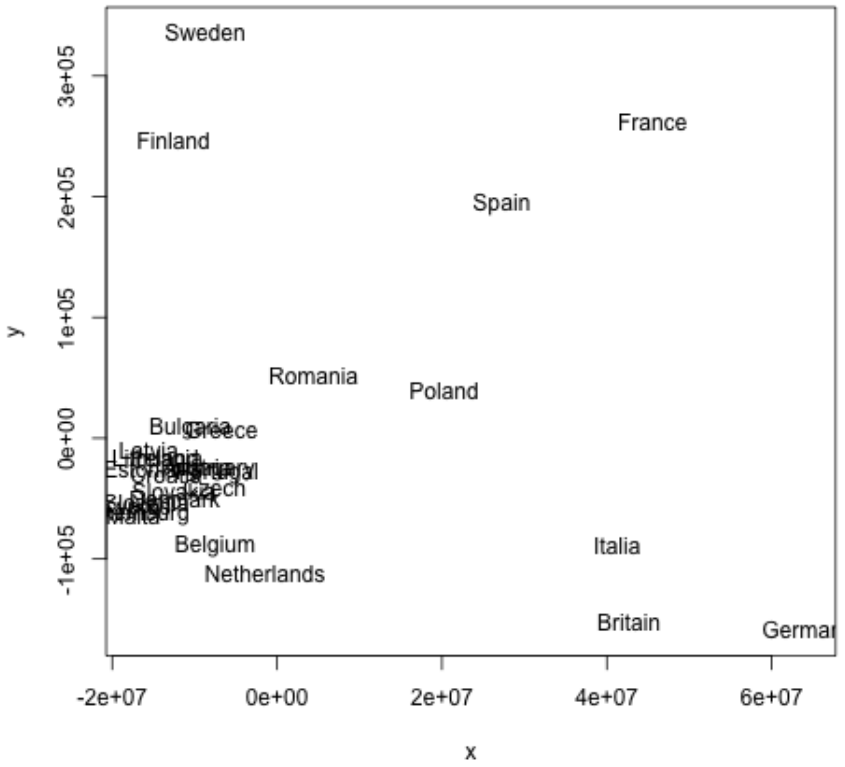
Relation Analysis using Polarity Words

- Cmdscale를 사용하여 2차원 공간상의 임의의 좌표점 계산하기

```
> data.4=cmdscale(data.3)
>
> x=data.4[,1]
>
> y=data.4[,2]
```

- 일반 plot을 사용하여 MDS그래프 생성

```
> png("total_plot.png")
>
> plot(x,y,type="n")
>
> text(x,y,labels=data.1[,1])
>
> dev.off()
null device
          1
```



Relation Analysis using Polarity Words

- 열이름 지정하기

```
> library(ggplot2)
>
> data.5=data.frame(data.4[,1],data.4[,2])
>
> colnames(data.5) <- c("X_axis","Y_axis")
```

- ggplot2를 사용하여 MDS그래프 생성

```
> png("total_ggplot.png")
>
> ggplot(data.5,aes(x=X_axis,y=Y_axis,colour=row.names(data.5)))+geom_point(alpha
=.5)+geom_text(aes(label=row.names(data.5)),size=4,vjust=2)+ggtitle("Relationship
beetwen Nations")
>
> dev.off()
null device
          1
```

Relation Analysis using Polarity Words

- ggplot2를 사용하여 MDS그래프 생성