Project_Bloomberg

FrannieK

19 June 2016

This project is comprised of two parts: i) text data preparation for Principal Component Analysis; ii) text analysis including Principal Component Analysis (PCA) through SPSS. The nature of text-type date requires certain cleansing and parsing processes in order to make the data appropriate for significant statistical analyses. Once the data is ready, the input is then loaded onto SPSS and ran through dimension-reduction techniques to identify associations and proximity between terms.

Preamble

I want to read as much news as possible but I seldom find time or ability to concentrate. I am also sometimes biased in my selection of articles based on the hot topics or on my existing knowledge scope. I wanted to expand my reading at the same time as not exhausting too much time. I hence constructed a method to extract key words from articles and map them based on relevance or association to capture the essence of a suite of articles.

Method

First, packages required are stated in the chunk below. "tm" is a useful package to treat text-type data. A number of articles have been saved in a central depository then read into R. Originally, the entire article is a one chunk but this is broken down into paragraphs (per element in a list) then treated to remove space, punctuation, grammatical insertions (articles, pronouns etc.). At this point the class of data is "Corpus" which is then converted into vectorised "list" for further parsing. Paragraphs are first broken into individual words but still grouped in *strsplit* then *unlist* merged all sub-lists to form one list per article.

Key Words

"Key" words are proxied by the n most frequent words in a selection of articles. N most frequent words are chosen from the entire collection of articles as an aggregate. Then relative frequencies are counted for each article (which later forms one observation in the final data output). If there were m number of articles, then m x n matrix is finally created as the the ultimate output. This is fed into SPSS in csv format for further statistical analyses.

SPSS

Rows (number of articles) does not have any meaning and all variables are represented in columns and the frequencies represente the occurence of key words in each article. In this case, Principle Component Analysis is most suitable as the metrics are numeric. (if categorical and two-way then use Correspondence Analysis).

Instruction for PCA: Load csv data with variables as columns -> verify -> Analyze -> Dimension Reduction -> Factor -> Descriptive: correlation matrix -> Extraction: Correlation matrix, (Unrotated) factor solution, scree plot -> Rotation: varimax, rotated solution, loading plots -> Ok -> if PCA plots comes out as 3D: click chart then edit then edit then select "Z" axis then property then variable then "Exclude".

Limitation / Evaluation

1. Want to include time stamp for time-series analysis and also for added factor in PCA. I can observe how key words and their intercorrelations change over time
2. Learn better how to interpret PCA output
3. Apply the results of PCA in trading for realise benefits of this project.

```
library(tm)
```

```
## Loading required package: NLP
```

```
library(knitr)

opts_chunk$set(comment = "", warning = FALSE, message = FALSE, echo = TRUE, tidy = TRUE, size="small", i
```

```
dir = "C:/Users/Huran/Desktop/Data_Project/Proj_Bloomberg"
setwd(dir)
set.seed(2016)

### repeat of data import ##################################################
cname = file.path(dir, "texts")  #
cname  #
dir(cname)  #
#
docs = Corpus(DirSource(cname))
docs = tm_map(docs, stemDocument)  # data cleansing to singularise, de-tense
docs = tm_map(docs, removeWords, stopwords("english"))  # data cleansing to remove common english words
docs = tm_map(docs, removePunctuation)  # data cleansing
docs = tm_map(docs, stripWhitespace)  #data cleansing to remove blank characters
# docs = tm_map(docs, removeNumbers) # remove numbers in docs
docs = tm_map(docs, removeWords, c("Bloomberg", "The", "A"))
for (i in seq(docs)) {
    docs[[i]] = gsub("Hong Kong", "Hong_Kong", docs[[i]])
}
inspect(docs)
class(docs)

# docs = docs[3:4] # if also captures back up files
```

```
library(stringr)
################ list-ify individual words ################################
text = vector("list", length(docs))  #cretes a series of list (can be of different length) in a vector
for (i in seq(docs)) {
    text[[i]] = strsplit(as.character(docs[[i]]), " ")  #parse text into individual strings but broken
    text[[i]] = unlist(text[[i]])  # merge into single list
    text[[i]] = text[[i]][nzchar(text[[i]])]  #only non-blank cells

}
summary(text)
save(text, file = "bloombergtext.RData")
```

```r
n = 50  #top n words in each list

all = unlist(text)
freq = sort(table(all), decreasing = TRUE)
# paste(text) #alt.1
key.word = names(head(freq, n))  # top n key words

word.count = lapply(key.word, function(x) data.frame(count = sapply(text, function(z) sum(str_count(z,
    x)))))  # get count of key words in each text vector

names(word.count) <- key.word
output = matrix(unlist(word.count), ncol = 50, byrow = FALSE)
dimnames(output) <- list(1:nrow(output), key.word)


### top n words by each article (not using for now) ###
table(text[[1]])
head(sort(table(text[[1]]), decreasing = TRUE), 20)


input = vector("list", length(text))
for (j in seq(text)) {
    input[[j]] = head(sort(table(text[[j]]), decreasing = TRUE), n)
}
#############################################################

write.csv(output, file = "Blmbgdata.csv", row.names = FALSE)  #save matrix in xlsx

# READY!(29.06.2016)
```

```
      Length Class  Mode
 [1,]  245   -none- character
 [2,]  175   -none- character
 [3,]  211   -none- character
 [4,]  315   -none- character
 [5,]    1   -none- character
 [6,] 1198   -none- character
 [7,]  301   -none- character
 [8,]  378   -none- character
 [9,]  276   -none- character
[10,]  568   -none- character
```

```
          12           13         1979         1982         2016
           1            1            1            1            1
         20s           23         2352          30s          370
           1            1            1            1            1
       37873          40s    42yearold    46yearold          587
           1            1            1            1            1
         674            8          Abe       Abenom       accept
           1            1            1            1            1
      accord       affect        allow       amount          ani
           1            1            6            1            1
        apan        April    architect           As      attempt
```

1	2	1	1	1
average	bad	Bank	benefit	biggest
1	2	1	1	1
BOJ	budget	budgets	chang	change
1	1	1	1	1
cheap	cite	come	conduct	control
1	1	2	1	1
corpor	cut	daili	day	declined
1	2	1	1	1
deflationari	deliveri	easiest	economi	economist
1	1	1	2	1
enjoy	experi	fail	famili	famous
1	1	1	1	1
first	flat	found	front	future
1	1	1	1	1
gains	go	gone	good	Governor
1	1	1	1	1
growth	hard	Haruhiko	Hayano	help
1	1	1	1	1
home	hours	I	improv	Inc
2	1	2	1	1
inflat	interview	It	Japan	Japanes
1	1	2	1	2
Katsunori	Kohei	Koya	Kuroda	Last
1	1	1	1	1
lender	less	level	life	like
1	1	1	1	1
longsuff	Ltd	lunch	man	may
1	1	3	2	1
meager	Meanwhile	men	Middleag	mindset
1	1	6	1	1
Minist	Miyamae	money	month	near
1	2	3	3	1
negoti	news	Nikko	Now	often
1	1	1	1	1
One	Oyama	past	people	per
1	1	1	1	1
percent	person	pocket	Pocket	popular
2	1	1	1	1
price	Prime	probabl	profits	program
1	1	1	1	1
prolong	prospect	record	reduc	reductions
1	1	1	1	1
reflat	reinflat	relief	result	reviv
1	1	1	1	1
rise	rose	said	salari	Salari
1	2	7	3	1
salarymen	save	saw	say	secondlowest
1	1	1	1	1
Secur	seen	set	sever	Shinsei
1	1	1	1	1
Shinzo	sign	sinc	SMBC	solid
1	1	1	1	1
spend	start	stood	store	strong

```
         6          1          1          1          1 
   struggl     survey      tepid      There      thing 
         1          5          1          1          1 
thirdlowest      three       toil   Tokyobas        two 
         1          1          1          1          1 
    Uniqlo       wage        way  Wednesday       went 
         1          1          1          1          1 
     While       will       wive      wives      women 
         1          1          1          1          1 
    worker    workers      worst       year      years 
         1          1          2          4          1 
       yen        You 
         4          1 

  said   allow     men   spend  survey    year     yen   lunch   money 
     7       6       6       6       5       4       4       3       3 
 month  salari   April     bad    come     cut economi    home       I 
     3       3       2       2       2       2       2       2       2 
    It Japanes 
     2       2 
```

reference: “https://deltadna.com/blog/text-mining-in-r-for-term-frequency/” “https://rstudio-pubs-static.s3.amazonaws.com/31867_8236987cf0a8444e962ccd2aec46d9c3.html#loading-texts"http://yihui.name/knitr/options/”