

Text_Mining_Reuters

Montse Figueiro

4 de julio de 2016

Preprocesamiento con el Paquete tm

Vamos a terminar construyendo una matriz de frecuencias, las filas son documentos, las entradas son el número de veces que ocurren, las columnas son términos, puedes estudiar clasificación, clustering, detección de temas, representación gráfica, nubes de palabras. Los documentos pueden ser novelas, estas matrices son bags of words. Es convertirlo en conteo de palabras, con eso destruyes el texto pero puedes hacer ciertas cosas. A veces te quedas con trigramas (trios de palabras) o n-gramas (n palabras consecutivas en el texto). Los trigramas son más útiles en el inglés (muy estructurado), porque todo está hecho para el inglés y el español es un poco diferente.

ngrams

Matrices de frecuencias:

Pasos a Seguir:

- tokenización: parte los textos en palabras. Las palabras que componen un texto, cuáles son los separadores posibles, hay un problema por ejemplo expresiones como “estar en los cerros de úbeda” es una expresión como una sola palabra. Nombres propios es una sola palabra “Tribunal de cuentas” es una sola palabra, esto exige tener un buen “ner” named entity recognition, que te permita detectar objetos sujetos cuyo nombre consta de varias palabras. Coges el BOE y quieres saber de qué se está hablando. Es más complicado de lo que parece. Mayúsculas separadas por un de, hay un diccionario de nombres propios y los encuentra...
- Eliminación palabras comunes (y, la, a), son demasiado comunes y no sirven para los análisis, hay listas por idiomas estas palabras no aportan nada, son frecuentes en todos los documentos, TF-IDF es una medida que da peso a los términos que aparecen frecuentemente pero que quieres que aparezcan no en todos los documentos sino también en un subconjunto de documentos. TF- term frequency IDF- inverse document frequency que aparezca en pocos documentos. quieres dar más peso a uno sobre otro, quieres calibrar qué parte es más interesante. cogemos texto, quitamos palabras comunes, tienen el TFIDF más alto.
- Lematización: buscar raíz de las palabras (casa puede ser de casa, de casar) el verbo puede tener más de 100 formas distintas en español, el español es muy flexible morfológicamente, una palabra suelta no sabes qué raíz tiene “casas”, es casa o casar? hay métodos basados en reglas, snowball. Otros basados en diccionarios, es una búsqueda directamente en diccionario, solo que te puede dar varias raíces. Basados en máquinas de estados finitos. puedes tener todos los verbos conjugados (esto lo hace el móvil cuando predice).

Snowball: lo usan mucho, es un algoritmo, te da una colección de raíz de palabras, en algunos casos coincide en otros no. Es muy bruto.

Si quieres contar el número de palabras en un discurso, con Snowball no queda bien, hay palabras que las utilizas con varios géneros, no las suma.

- sinónimos: cuando escribes tratas de no repetir palabras, te gustaría deshacer eso, hay que utilizar diccionarios.

Con esto ya tenemos la matriz de frecuencias.

Hoy te casas, hoy es adverbio, te es pronombre y casas puede ser nombre overbo, te calcula las probabilidades de que sea una opción u otra. Usando modelos de Markov

Librerías

```
library(tm)
```

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

```
library(RColorBrewer)
library(wordcloud)
```

Corpus Sources and Readers:

```
getSources()
```

```
## [1] "DataframeSource" "DirSource"          "URISource"          "VectorSource"
## [5] "XMLSource"       "ZipSource"
```

```
getReaders()
```

```
## [1] "readDOC"          "readPDF"
## [3] "readPlain"        "readRCV1"
## [5] "readRCV1asPlain"  "readReut21578XML"
## [7] "readReut21578XMLasPlain" "readTabular"
## [9] "readTagged"       "readXML"
```

Data Reuters

```
data("acq")
acq[[1]]
```

```
## <<PlainTextDocument>>
## Metadata: 15
## Content: chars: 1287
```

```
ruta<- system.file("texts", "acq", package = "tm")
ruta
```

```
## [1] "D:/Users/msi/Documents/R/win-library/3.3/tm/texts/acq"
```

```
reuters <- VCorpus(DirSource(ruta),
                  readerControl = list(reader = readReut21578XMLasPlain))
```

```
reuters[[1]]
```

```
## <<PlainTextDocument>>
## Metadata: 16
## Content: chars: 1287
```

```
inspect(reuters[1])
```

```
## <<VCorpus>>
## Metadata: corpus specific: 0, document level (indexed): 0
## Content: documents: 1
##
## [[1]]
## <<PlainTextDocument>>
## Metadata: 16
## Content: chars: 1287
```

```
str(reuters[1])
```

```
## List of 1
## $ 10:List of 2
## ..$ content: chr "Computer Terminal Systems Inc said\nit has completed the sale of 200,000 shares of its common\n
## ..$ meta :List of 16
## .. ..$ author : chr(0)
## .. ..$ timestamp: POSIXlt[1:1], format: NA
## .. ..$ description : chr ""
## .. ..$ heading : chr "COMPUTER TERMINAL SYSTEMS <CPML> COMPLETES SALE"
## .. ..$ id : chr "10"
## .. ..$ language : chr "en"
## .. ..$ origin : chr "Reuters-21578 XML"
## .. ..$ topics : chr "YES"
## .. ..$ lewissplit : chr "TRAIN"
## .. ..$ cgisplit : chr "TRAINING-SET"
## .. ..$ oldid : chr "5553"
## .. ..$ topics_cat : chr "acq"
## .. ..$ places : chr "usa"
## .. ..$ people : chr(0)
## .. ..$ orgs : chr(0)
## .. ..$ exchanges : chr(0)
## .. ..- attr(*, "class")= chr "TextDocumentMeta"
## ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
## - attr(*, "class")= chr [1:2] "VCorpus" "Corpus"
```

```
reuters[[1]]$content
```

```
## [1] "Computer Terminal Systems Inc said\nit has completed the sale of 200,000 shares of its common\n"
```

Un Corpus es una lista de documentos; cada documento tiene el texto y un conjunto de metadatos (que no usaremos)

Transformaciones: tm_map aplica una función a cada documento

```
reuters <- tm_map(reuters, stripWhitespace) # quitar los espacios en blanco que están de más sobre cada  
reuters <- tm_map(reuters, content_transformer(tolower)) #tolower hay que meterla dentro sino no funciona  
#pasa todas las palabras a minúsculas  
reuters <- tm_map(reuters, removePunctuation) #elimina comas y puntos  
reuters <- tm_map(reuters, removeWords, stopwords("en")) #quitamos palabras stopwords, trae una lista, so  
#comunes sin importancia en el análisis.
```

```
reuters[[1]]$content
```

```
## [1] "computer terminal systems inc said    completed   sale   200000 shares    common stock  warrants a
```

```
class(reuters)
```

```
## [1] "VCorpus" "Corpus"
```

Raíz de las palabras con Snowballc

```
library(SnowballC)  
Snowreuters <- tm_map(reuters[1:10], stemDocument)  
Snowreuters[[1]]$content
```

```
## [1] "comput termin system inc said    complet   sale   200000 share    common stock  warrant  acquir  ad
```

WordCloud reuters

```
wordcloud(reuters,scale=c(5,0.5),max.words=100,random.order=FALSE,rot.per=0.35,use.r.layout=FALSE,  
          colors=brewer.pal(8, "Dark2"))
```


Frequency Words

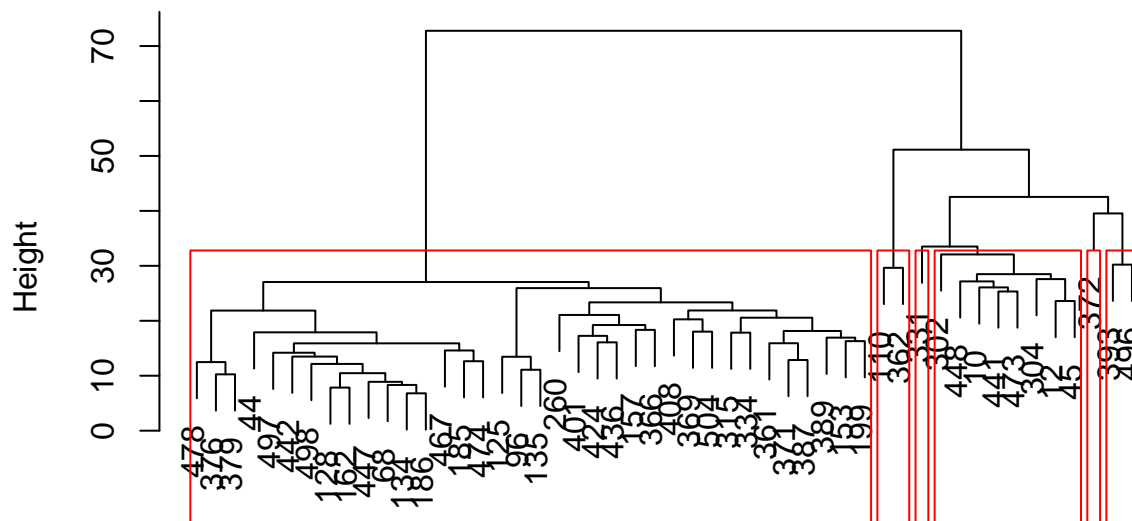
```
matrixreuters <- as.matrix(matrix)
frequency <- colSums(matrixreuters)
frequency <- sort(frequency, decreasing=TRUE)
head(frequency,10)
```

```
##      said      dlrs      pct      mln company      inc      shares      reuter      stock
##      186       100       70       65       63       53       52       50       46
##      will
##       35
```

Frequency words removing sparse Terms (this terms appears in a few documents)

```
matrixreuters2<- removeSparseTerms(matrix, sparse = 0.95)
m2 <- as.matrix(matrixreuters2)
distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method = "ward.D")
plot(fit)
rect.hclust(fit, k = 6) # cut tree into 6 clusters
```

Cluster Dendrogram



distMatrix
hclust (*, "ward.D")

```
frequency2 <- colSums(m2)
frequency2 <- sort(frequency2, decreasing=TRUE)
head(frequency2,10)
```

```
##      said      dlrs      pct      mln company      inc  shares  reuter  stock
##      186      100      70      65      63      53      52      50      46
##      will
##      35
```

```
inspect(matrixreuters2[1:10,1:10])
```

```
## <<DocumentTermMatrix (documents: 10, terms: 10)>>
## Non-/sparse entries: 11/89
## Sparsity          : 89%
## Maximal term length: 12
## Weighting          : term frequency (tf)
##
##      Terms
## Docs  125 1985 1986 1987 200 acquire acquired acquisition acquisitions
##  10    1   0   0   0   0      1       0           0           0
##  12    0   0   1   0   0      0       0           0           3
##  44    0   0   0   0   0      0       0           0           0
##  45    0   0   1   0   0      0       1           0           0
##  68    0   0   0   0   0      0       1           0           0
##  96    0   0   0   0   0      0       0           0           0
## 110    0   1   0   0   1      0       0           0           1
## 125    0   0   0   0   0      0       0           0           0
## 128    0   0   0   0   0      0       0           0           0
## 134    0   0   0   0   0      0       0           0           0
##      Terms
## Docs  added
##  10      0
##  12      0
##  44      0
##  45      0
##  68      0
##  96      0
## 110      1
## 125      0
## 128      0
## 134      0
```

TP-IDF

Normaliza le quita importancia a las palabras que aparecen repetidas en muchos documentos. Le quita peso.

```
reuters.norm <- weightTfIdf(matrix)
inspect(reuters.norm[1:10,1:10])
```

```
## <<DocumentTermMatrix (documents: 10, terms: 10)>>
## Non-/sparse entries: 2/98
```

```
## Sparsity          : 98%
## Maximal term length: 6
## Weighting         : term frequency - inverse document frequency (normalized) (tf-idf)
##
##      Terms
## Docs  05165 0523      100 10000 100000      101 105 1078 110 1100
##  10      0      0 0.0000000      0      0 0.00000000      0      0      0      0
##  12      0      0 0.0000000      0      0 0.00000000      0      0      0      0
##  44      0      0 0.0000000      0      0 0.00000000      0      0      0      0
##  45      0      0 0.0000000      0      0 0.02577103      0      0      0      0
##  68      0      0 0.0000000      0      0 0.00000000      0      0      0      0
##  96      0      0 0.1132648      0      0 0.00000000      0      0      0      0
## 110      0      0 0.0000000      0      0 0.00000000      0      0      0      0
## 125      0      0 0.0000000      0      0 0.00000000      0      0      0      0
## 128      0      0 0.0000000      0      0 0.00000000      0      0      0      0
## 134      0      0 0.0000000      0      0 0.00000000      0      0      0      0
```

```
reuters.norm.matrix <- as.matrix(reuters.norm)
frequency.norm <- colSums(reuters.norm.matrix)
frequency.norm <- sort(frequency.norm, decreasing=TRUE)
head(frequency.norm)
```

```
##  shares  liebert    dlrs    rmj    corp    mln
## 0.7854376 0.7054820 0.6901116 0.6870388 0.6542134 0.6469322
```

La matriz de resultados normalizados la podemos pasar a data.frame

```
Res <- as.data.frame(inspect(reuters.norm[,c("said", "company"))))
```

```
## <<DocumentTermMatrix (documents: 50, terms: 2)>>
## Non-/sparse entries: 27/73
## Sparsity          : 73%
## Maximal term length: 7
## Weighting         : term frequency - inverse document frequency (normalized) (tf-idf)
##
##      Terms
## Docs  said    company
##  10      0 0.027352883
##  12      0 0.011545048
##  44      0 0.000000000
##  45      0 0.016236871
##  68      0 0.000000000
##  96      0 0.000000000
## 110      0 0.012996618
## 125      0 0.014573257
## 128      0 0.000000000
## 134      0 0.037040362
## 135      0 0.000000000
## 153      0 0.015595942
## 157      0 0.029965237
## 162      0 0.000000000
```



```
## 185 0 0.000000000
## 186 0 0.049387149
## 199 0 0.000000000
## 260 0 0.019325406
## 302 0 0.007377334
## 304 0 0.000000000
## 315 0 0.000000000
## 331 0 0.004857752
## 334 0 0.000000000
## 361 0 0.000000000
## 362 0 0.005437117
## 366 0 0.009988412
## 369 0 0.014110614
## 371 0 0.013268189
## 372 0 0.020073486
## 376 0 0.040407668
## 379 0 0.000000000
## 387 0 0.000000000
## 389 0 0.000000000
## 393 0 0.011696956
## 401 0 0.021949844
## 408 0 0.008386497
## 424 0 0.000000000
## 436 0 0.000000000
## 441 0 0.010974922
## 442 0 0.000000000
## 447 0 0.000000000
## 448 0 0.024557146
## 467 0 0.000000000
## 473 0 0.025159491
## 474 0 0.018142218
## 478 0 0.000000000
## 496 0 0.017545435
## 497 0 0.057976219
## 498 0 0.000000000
## 504 0 0.000000000
```

```
Res[, "company"]
```

```
## [1] 0.027352883 0.011545048 0.000000000 0.016236871 0.000000000
## [6] 0.000000000 0.012996618 0.014573257 0.000000000 0.037040362
## [11] 0.000000000 0.015595942 0.029965237 0.000000000 0.000000000
## [16] 0.049387149 0.000000000 0.019325406 0.007377334 0.000000000
## [21] 0.000000000 0.004857752 0.000000000 0.000000000 0.005437117
## [26] 0.009988412 0.014110614 0.013268189 0.020073486 0.040407668
## [31] 0.000000000 0.000000000 0.000000000 0.011696956 0.021949844
## [36] 0.008386497 0.000000000 0.000000000 0.010974922 0.000000000
## [41] 0.000000000 0.024557146 0.000000000 0.025159491 0.018142218
## [46] 0.000000000 0.017545435 0.057976219 0.000000000 0.000000000
```

Words correlation

```
findAssocs(matrix, "dlrs", 0.6)
```

```
## $dlrs
##      least      valued      rivals      unless
##      0.84      0.83      0.82      0.82
##      cash      takeover      provide      mln
##      0.81      0.78      0.77      0.74
##      revised      118      150      175
##      0.73      0.72      0.72      0.72
##      195      2275      295      3850
##      0.72      0.72      0.72      0.72
##      475      592      6881800      agreements
##      0.72      0.72      0.72      0.72
##      besides      bidding      bids      chain
##      0.72      0.72      0.72      0.72
##      commitment      confident      confidentiality      consist
##      0.72      0.72      0.72      0.72
##      consisting      contains      contribution      dedham
##      0.72      0.72      0.72      0.72
##      disclose      documents      drawn      face
##      0.72      0.72      0.72      0.72
##      fenner      formal      groups      half
##      0.72      0.72      0.72      0.72
##      information      keep      leads      limited
##      0.72      0.72      0.72      0.72
##      lynch      massbased      merrill      monthlong
##      0.72      0.72      0.72      0.72
##      newly      onefifth      operator      pierce
##      0.72      0.72      0.72      0.72
##      portion      purchases      purpose      records
##      0.72      0.72      0.72      0.72
##      redstone      redstones      sec      secret
##      0.72      0.72      0.72      0.72
##      separate      set      smith      submitted
##      0.72      0.72      0.72      0.72
##      sumner      sweeten      sweetened      syndicate
##      0.72      0.72      0.72      0.72
##      theater      toward      underwrite      underwriting
##      0.72      0.72      0.72      0.72
##      vying      war      financing      raise
##      0.72      0.72      0.70      0.70
##      committed      inc      share      4050
##      0.68      0.68      0.68      0.65
##      750      called      committee      eight
##      0.65      0.65      0.65      0.65
##      viacom      viacoms      later      proposed
##      0.65      0.65      0.64      0.64
##      two      contribute      offer      provided
##      0.63      0.62      0.62      0.62
##      earlier
##      0.60
```

```
findAssocs(matrix, "said", 0.6)
```

```
## $said
##      company      analysts      part      stock
##      0.75      0.74      0.72      0.70
##      316      aftertax      brothers      chairmen
##      0.69      0.69      0.69      0.69
##      considered      contributed      created      divisions
##      0.69      0.69      0.69      0.69
##      eckenfelder      expand      express      got
##      0.69      0.69      0.69      0.69
##      highly      internal      lane      larry
##      0.69      0.69      0.69      0.69
##      lehman      move      place      positions
##      0.69      0.69      0.69      0.69
## prudentialbach      remained      rumors      selling
##      0.69      0.69      0.69      0.69
##      sense      silent      spinoff      unlikely
##      0.69      0.69      0.69      0.69
##      vacant      shearson      american      reflect
##      0.69      0.68      0.66      0.66
##      operating      services      fully      market
##      0.65      0.64      0.63      0.63
##      believe      chief      however      officer
##      0.62      0.62      0.62      0.62
##      several      shearsons      speculated      spinning
##      0.62      0.62      0.62      0.62
##      strong      future      also
##      0.62      0.61      0.60
```

```
findAssocs(matrix, "pct", 0.6)
```

```
## $pct
##      stake      increased      interests      rights      option
##      0.74      0.69      0.69      0.66      0.63
##      cost      key      126      148      1984
##      0.62      0.62      0.61      0.61      0.61
##      341      400      424      455      494
##      0.61      0.61      0.61      0.61      0.61
##      activities      alfa      alfs      amounted      arms
##      0.61      0.61      0.61      0.61      0.61
##      aseas      asts      atlas      back      building
##      0.61      0.61      0.61      0.61      0.61
##      buyers      concentrating      copco      core      crowns
##      0.61      0.61      0.61      0.61      0.61
##      defend      diary      empire      erik      expensive
##      0.61      0.61      0.61      0.61      0.61
##      farflung      financier      forced      foreign      forvaltnings
##      0.61      0.61      0.61      0.61      0.61
##      fought      frederik      free      fringe      heart
##      0.61      0.61      0.61      0.61      0.61
##      incentive      industrier      investors      koppabergs      laval
```

##	0.61	0.61	0.61	0.61	0.61
##	left	londonbased	lundberg	managed	match
##	0.61	0.61	0.61	0.61	0.61
##	matchs	nobel	nobl	originally	ousted
##	0.61	0.61	0.61	0.61	0.61
##	outside	penser	predators	prevent	protect
##	0.61	0.61	0.61	0.61	0.61
##	providentia	raid	restricted	secure	skanska
##	0.61	0.61	0.61	0.61	0.61
##	skbs	skf	skfr	skps	small
##	0.61	0.61	0.61	0.61	0.61
##	smbs	stora	swedens	swedish	taken
##	0.61	0.61	0.61	0.61	0.61
##	thre	tycoon	undertaken	volv	volvo
##	0.61	0.61	0.61	0.61	0.61
##	wallenberg	wallenbergs	wrested	zurichbased	voting
##	0.61	0.61	0.61	0.61	0.60

```
findAssocs(matrix, "mln", 0.6)
```

```
## $mln
## dlrs
## 0.74
```