Documentación Proyecto final

Felipe Gerard, Omar Díaz, Andrés Villaseñor

21 de mayo de 2015

Limpieza de la colección de abstracts

0. Descripción del proceso

La limpieza de los datos consiste en los siguientes pasos:

1. Conversión de archivos de texto a JSON estructurado en *bash* (`abstract2json.sh`):

```
  {
"1":{
      "Titulo":{"Ejemplo de título"},
      "Fecha":{"2013-01-02"},
      "Abstract":{"Ejemplo de abstract"},
      ...
      },
  "2":{
      "Titulo":{"Ejemplo de otro título"},
      "Fecha":{"2003-03-02"},
      "Abstract":{"Ejemplo de otro abstract"},
      ...
      },
  ...
  }
```

2. Conversión de JSON estructurado a data.frame de R para el análisis en *R* (`json2dataframe_abstracts.R`):

3. **Texto –> JSON**

El primer paso es sustituir los signos de puntuación y los saltos de línea en caracteres manejables. El fin de esto es que no afecten los JSON ni generen resultados raros. Usando los comandos `tr` y `sed` convertimos los saltos de línea en " ", los puntos por"", etc.

Ya con la puntuación resuelta, pasamos los txt a JSON. Los archivos originales ya venían en un formato razonable similar a JSON, "nombre_de_atributo:descripción", así que lo pudimos convertir sin problema agregando las comillas pertinentes, etc. Hubo que tener cuidado con el encoding de los textos y con el hecho de que el formato no era 100% formal. Para esta parte utilizamos sobre todo `sed` y `awk`.

2. JSON –> data.frame

Primero nos aseguramos de que el formato sea legible en *R*. Cuando no, utilizamos *Python* para limpiar el formato. Una vez habiendo leído la lista a *R* con la librería `jsonlite`, convertimos la lista recursiva a `data.frame`. Para ahorrar tiempo, utilizamos la función `mclapply` del paquete `parallel`. Finalmente, regresamos la puntuación a su forma original para que el texto recuperado sea legible.

Ejemplo

1. Texto original:

```
Title       : RFLP Patterns as a Measure of Diversity in Small Populations
Type        : Award
NSF Org     : MCB
Latest
Amendment
Date        : May 31,  1990
File        : a9000031

Award Number: 9000031
Award Instr.: Standard Grant
Prgm Manager: Maryanna P. Henkart
MCB  DIV OF MOLECULAR AND CELLULAR BIOSCIENCE
BIO  DIRECT FOR BIOLOGICAL SCIENCES
    Start Date  : June 1,  1990
Expires     : May 31,  1994        (Estimated)
    Expected
    Total Amt.  : $300000             (Estimated)
Investigator: Marcia M. Miller mamiller@coh.org  (Principal Investigator current)
    Sponsor     : Beckman Res Inst Cty Hope
    1500 E. Duarte Road
    Duarte, CA  910103000    /   -

    NSF Program : 1114      CELL BIOLOGY
    Fld Applictn: 0000099   Other Applications NEC
    61        Life Science Biological
    Program Ref : 9285,
    Abstract    :

    Studies of chickens have provided serological and nucleic acid
    probes useful in defining the major histocompatibility complex
    (MHC) in other avian species.  Methods used in detecting genetic
    diversity at loci within the MHC of chickens and mammals will be
    applied to determining the extent of MHC polymorphism within
    small populations of ring-necked pheasants, wild turkeys, cranes,
    Andean condors and other species.  The knowledge and expertise
    gained from working with the MHC of the chicken should make for
    rapid progress in defining the polymorphism of the MHC in these
    species and in detecting the polymorphism of MHC gene pool within
    small wild and captive populations of these birds.

    Genes within the major histocompatibility complex (MHC) are known
    to encode molecules that provide the context for recognition of
    foreign antigens by the immune system.  Whether a given animal is
    able to mount an immune response to the challenge of a pathogen
    is determined, in part, by the allelic makeup of its MHC.  In
    many species, an unusually high degree of polymorphism is
    maintained at multiple loci within the MHC in freely breeding
    populations.  The allelic pool within a population presumably
    provides diversity upon which to draw in the face of
```

```
    environmental challenge.  The objective of the proposed research
    is to extend ongoing studies of the MHC of domesticated fowl to
    include avian species experiencing severe reduction in population
    size.  Knowledge of the MHC gene pool within populations and of
    the haplotypes of individual animals may be useful in the
    husbandry of species requiring intervention for their
    preservation.
```

2. JSON

```
"51758":{
    "Title":"RFLP Patterns as a Measure of Diversity in Small Populations",
    "Date":"May 31<coma> 1990",
    "Award Number":"9000031",
    "Investigator":"Marcia M<punto> Miller mamillercoh<punto>org <abre_parent>Principal
    Investigator current<cierra_parent>",
    "Sponsor":"Beckman Res Inst Cty Hope<br> 1500 E<punto> Duarte Road<br> Duarte<coma>
    CA 910103000 <diagonal> <guion><br>",
    "Fld Applictn":"0000099 Other Applications NEC <br> 61 Life Science Biological",
    "Abstract":"<br> <br> Studies of chickens have provided serological and nucleic
    acid <br> probes useful in defining the major histocompatibility complex <br>
    <abre_parent> MHC<cierra_parent> in other avian species<punto> Methods used in
    detecting genetic <br> diversity at loci within the MHC of chickens and mammals will
    be <br> applied to determining the extent of MHC polymorphism within <br> small
    populations of ring<guion>necked pheasants <coma> wild turkeys<coma> cranes<coma> <br>
    Andean condors and other species<punto> The knowledge and expertise <br> gained from
    working with the MHC of the chicken should make for <br> rapid progress in defining
    the polymorphism of the MHC in these <br> species and in detecting the polymorphism
    of MHC gene pool within <br> small wild and captive populations of
    these birds<punto> <br> <br> Genes within the major histocompatibility complex
    <abre_parent> MHC<cierra_parent> are known <br> to encode molecules that provide
    the context for recognition of <br> foreign antigens by the immune system<punto>
    Whether a given animal is <br> able to mount an immune response to the challenge
    of a pathogen <br> is determined<coma> in part<coma> by the allelic makeup of its
    MHC<punto> In <br> many species<coma> an unusually high degree of polymorphism
    is <br> maintained at multiple loci within the MHC in freely breeding <br>
    populations<punto> The allelic pool within a population presumably <br>
    provides diversity upon  which to draw in the face of <br>
    environmental challenge<punto> The objective of the proposed research <br>
    is to extend ongoing studies of the MHC of domesticated fowl to <br>
    include avian species experiencing severe reduction in population <br>
    size<punto> Knowledge ofthe MHC gene pool within populations and of <br>
    the haplotypes of individual animals may be useful in the <br>
    husbandry of species requiring intervention for their <br>
    preservation<punto><br>"
}
```

3. data.frame

Tiene las columnas del JSON anterior pero en versión y la puntuación en formato normal. No incluimos el ejemplo porque el formato no es práctico.

4. Implementación del modelo

Una vez con los datos limpios como se mencionó anteriormente, procedemos a implementar el modelo, para esto tenemos lo siguiente.

```
#Las librerías utilizadas son las siguientes:
library(Matrix)
library(dplyr)
library(tm)
library(slam)
library(Rstem)
library(ggplot2)
library(wordcloud)
library(knitr)

#carga de datos
#load('abstracts_clean.Rdata')
abstracts2 <- as.data.frame(abstracts2)


# filtramos cosas feas en abstract y title, es decir nos quedamos
# con los datos 100% limpios
d <- abstracts2 %>%
  filter(grepl('Presidential Awardee',Title)=='FALSE') %>% #815
  filter(grepl('Not Available',Abstract)=='FALSE') %>% #1267
  filter(Title != '') %>% #8
  filter(Abstract != '' ) %>% #2180
  filter(grepl('----------------------------------/
               ----------------------------------------',Abstract)=='FALSE')

#filtrado de información y última limpieza antes de crear el corpus
d1 <- d %>%
  mutate(id=row_number()) %>%
  select(id,Title,Abstract)
v <- filter(d1,grepl('S u m',Abstract)=='TRUE')
```

```
# Creamos el corpus y limpiamos caracteres especiales
corpus.frases <- Corpus(VectorSource(d$Abstract))

corp.1 <- tm_map(corpus.frases,function(x){
  c1 <- gsub('R o o t E n t<br> r y','Root Entry',x)
  c2 <- gsub('C o m p O b j','Comp Obj',c1)
  c3 <- gsub('S u m m a r y I n f o r m a t i o n','Summary Information',c2)
  c4 <- gsub('<br> b <br>','',c3)
  c5 <- gsub('W o r d D o c u m e n t','Word Document',c4)
  c6 <- gsub('O b j e c t P o o l','Object Pool',c5)
  c7 <- gsub('[-]|<br>',' ',c6)
  gsub('[()]|[.,;:`"*#&/><]|[\\\']|[]\\[]','',c7)
})

corp.2 <- tm_map(corp.1,removeWords,stopwords("english"))
corp.2 <- tm_map(corp.2, function(x) stripWhitespace(x) %>% tolower)
corp.2 <- tm_map(corp.2,function(x){
  z <- strsplit(x, " +")[[1]]
```

```
  z.stem <- wordStem(z, language="english")
  PlainTextDocument(paste(z.stem, collapse=" "))
})
```

Posteriormente creamos la matriz de términos-documentos y tomamos los pesos utilizano como criterio **"ntc"** ya que observamos que es el que mejor resultado genera.

```
#creamos la matriz terminos documentos
tdm.1 <- TermDocumentMatrix(corp.2, control=list(wordLengths=c(3, Inf)))
colnames(tdm.1) <- seq(1,tdm.1$ncol)

#eliminamos los documentos que no tienen terminos (empty docs)
#a través de la suma de las columnas
idx_sum <- as.numeric(as.matrix(rollup(tdm.1, 1, na.rm=TRUE, FUN = sum)))
tdm_new <- tdm.1[,idx_sum>0]

#actualizamos los pesos
tdm.2 <- weightSMART(tdm_new, spec = 'ntc')
#revisamos la normalización de los pesos
head(sort(vec.1 <- as.matrix(tdm.2[,500]),dec=T))
```

```
## [1] 2.3091553 2.0706309 1.3340512 1.2277843 1.1748767 0.9964995
```

Ya con el modelo listo, necesitamos hacer queries a la *"base de datos"* por lo que tenemos que homogenizar el query con los datos, es deicr tenemos que transformarlos a una matriz de términos documentos de la siguiente forma.

```
query <- 'The main objective of this proposal is better understanding of underlying atomic<br> and  mol
#limpieza del query
query.l <- Corpus(VectorSource(query))
q.1 <- tm_map(query.l,function(x){
  q1 <- gsub('[-]|<br>',' ',x)
  gsub('[()]|[.,;:`"*#&/><]|[\\\']|[]\\[]','',q1)
})
q.2 <- tm_map(q.1,removeWords,stopwords("english"))
q.2 <- tm_map(q.2, function(x) stripWhitespace(x) %>% tolower)
q.2 <- tm_map(q.2,function(x){
  z <- strsplit(x, " +")[[1]]
  z.stem <- wordStem(z, language="english")
  PlainTextDocument(paste(z.stem, collapse=" "))
})
query.limp <- q.2
```

Ahora tenemos que multiplicar la base (la matriz rala de todos los términos documentos) por la matriz obtenida anteriormente. Esto ya que nos permite quedarnos con los términos que aportan más al score TF-IDF y nos permiten medir la distancia en este caso en particular utilizando la **"distancia coseno"**, ya que el producto punto dividido entre la logitud de estas matrices es precisamente el tamaño de la intersección entre estos dos *conjuntos*. Por lo tanto este calculo nos permite conocer el coseno del angulo entre estos dos *"vectores"*

```
# Multiplicacion query * tdm
mat.1 <- sparseMatrix(i=tdm.2$i, j=tdm.2$j, x = tdm.2$v)
dictionary <- tdm.2$dimnames$Terms
query.vec.1 <- TermDocumentMatrix(query.limp,
                                  control = list(dictionary = dictionary,
                                                 wordLengths=c(1, Inf)))

#normalizar con ntc el query
query.vec.norm <- as.matrix(query.vec.1)/sqrt(sum(query.vec.1^2))

aa <- t(mat.1)%*%query.vec.norm
```

Ahora vislualizamos los 15 resultados que más aportarón información al modelo

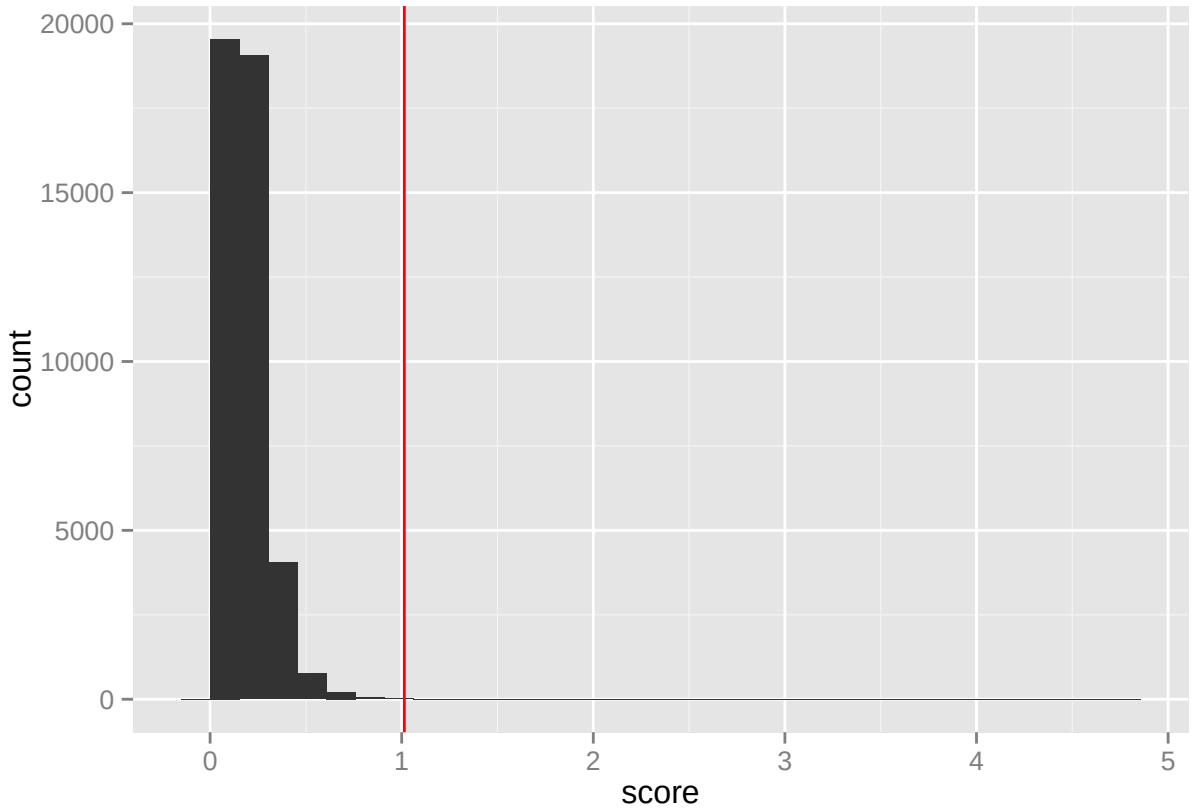
```
idx_top <- order(aa, decreasing=T)
out <- d[idx_top,] %>%
  select(Title, Date, Sponsor, Abstract) %>%
  cbind(score = sort(aa,decreasing = T)) %>%
  filter(score > 0)
res <- out %>% head(15)
res$Abstract <- gsub('<br>','',res$Abstract)
res$Title <- gsub('<br>','',res$Title)
res$Sponsor <- gsub('<br>','',res$Sponsor)

kable(res$Title)
```

Microstructural Design of Thio-Sol-Gel Derived Titanium Disulfide Particles
Tectonic Setting, Depth of Emplacement, and Unroofing History of the Middle Proterozoic Wolf River Batholith and Assoc
SGER: Phylogenetic and Indigeneity Implications of MALDI-MS Determined Osteocalcin Protein Sequences
Dissertation Research: Expression and Inheritance of a Life History Polymorphism
Anthropological Field Research on the Giant Lemurs and Associated Fauna of Madagascar
Mechanism of Sphingomyelinase Aativation by the Low Affinity Neurotrophin Receptor
Cross Section Measurements for Excitation and Ionization of Atmospheric Species
U.S.-Mongolia Cooperative Research: Satellite Telephone Communication System for Biodiversity Studies in Mongolia
A Multidisciplinary Control System Laboratory
Small Grant for Exploratory Research: Effect of Confinement on Reaction Equilibrium in Porous Materials
Collaborative Research: Interactions of Heavy Metals With Biofilm-Coated Mineral Surfaces
Vertex Algebras, S-duality Conjecture, and Counting Plane Curves
Design and Implementation of a Database System for Identified Neurons, Development, and Genetics of Zebrafish
Acquisition of an Ultra High Field Spectrometer
Structure-Function Studies of Lipid Binding Proteins

Observamos el histograma de discriminación, el cual nos da una idea gráfica e intuitiva de como es que estamos haciendo nuestro criterio de selección, el cual se encuentra a continuación.

```
m <- data.frame(min=min(res$score))
# Estadísticas. Son sobresalientes las palabras que mostramos?
ggplot() +
  geom_bar(data=out, mapping=aes(x=score)) +
  geom_vline(data=res, aes(xintercept=min(score)), color='red')
```



A la derecha de la línea roja, se marca la cantidad de documentos estamos considerando para hacer la recuperación de información. Si la cantidad de documentos a la derecha de la linea roja es muy grande quiere deicr que no estamos logrando una buena discriminación, de lo contrario tenemos que hay pocos documentos que nos aportan una ganancia de información, es decir, tenemos que la información es más valiosa y por lo tanto el resultado.

A continuación, vemos como obtenemos de forma particular las palabras que más contribuyen a nuestro modelo.

```
best <- function(nmatch = 3, nterm = 5){
  v.q <- query.vec.norm
  outlist <- list()
  for(i in 1:nmatch){
    #colnames(tdm.2)[idx_top[i]]
    #idx_top[nmatch]
    v.j <- mat.1[,idx_top[i]]
    v <- v.j*v.q
    #length(v)
    top_contrib <- order(v, decreasing = T)
    outlist[[i]] <- data.frame(term=dictionary[top_contrib[1:nterm]], # tdm.2$dimnames$Terms[top_contri
                               score_contrib=v[top_contrib[1:nterm]], stringsAsFactors=F) %>%
      filter(score_contrib > 0) %>%
      data.frame(rank = i, match = colnames(tdm.2)[idx_top[i]], total_score = sum(v), stringsAsFactors =
  }
  rbind_all(outlist)[c(3,4,5,1,2)] %>%
    group_by(term) %>%
    summarise(contrib=sum(score_contrib)) %>%
    arrange(desc(contrib))
}
```

```
best <- best(nmatch = 15, nterm = nrow(unique(as.data.frame(strsplit(as.character(query[[1]]),"  ")[[1]]
kable(head(best,15))
```

term	contrib
microstructur	9.4497394
properti	1.6682831
electrochem	0.9320445
evolut	0.6434028
size	0.5427946
process	0.4425614
object	0.2863774
charg	0.2495783
the	0.2288868
synthes	0.2053313
thermal	0.1990660
shape	0.1938246
investig	0.1742765
powder	0.1680078
orient	0.1665611

Finalmente el wordcloud para una visualización mas amigable.

```
wordcloud(best$term,best$contrib,
          scale=c(5,.7),
          min.freq=0.1,
          ordered.colors=T,
          colors=colorRampPalette(brewer.pal(9,"Set1"))(nrow(best)))
```