NLP

El Lazarillo de Tormes

MARIA TERESA ROMO GALLEGO
30-1-2022

Content

1. Document overview

In this assignment we will explore the tendency and frequency of the words found in a book, following a few points of view:

- First half vs second half
- Chapter vs chapter

The book in question will be El Lazarillo de Tormes, as found in:

https://www.gutenberg.org/cache/epub/320/pg320.txt

First, we start by loading the content of the URL as our input file, ensuring the encoding is correct. Since we are going to face a text in Spanish, in order to avoid weird characters appearing in it, we will choose UTF-8. Originally, I chose a file set in ISO-8859-1, however, since there is one already in UTF-8 available, we won't worry about reencoding it.

```
urlLazarillo <- "https://www.gutenberg.org/cache/epub/320/pg320.txt"
lines <- readLines(urlLazarillo, encoding = "UTF-8")
```

Let's take a look at the beginning and end of the file.

```
The Project Gutenberg EBook of La vida de Lazarillo de tormes y de sus
fortunas y adversidades, by Unknown

This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever.  You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.org


Title: La vida de Lazarillo de tormes y de sus fortunas y adversidades

Author: Unknown

Posting Date: March 18, 2012 [EBook #320]
Release Date: September, 1995

Language: Spanish


*** START OF THIS PROJECT GUTENBERG EBOOK LA VIDA DE LAZARILLO DE ***



End of the Project Gutenberg EBook of La vida de Lazarillo de tormes y de
sus fortunas y adversidades, by Unknown

*** END OF THIS PROJECT GUTENBERG EBOOK LA VIDA DE LAZARILLO DE ***
```

As we can see, just like in the first Hands On example, this file is also delimited by a quote between ***. We can figure out where the actual book begins and ends searching for this pattern:

```
> grep(pattern = "***", lines, fixed = TRUE)
[1]   20 2146 2148 2176
```

Lines 1 to 20 show us information on the document we're visualizing, so we will start from line 21. From there we will select every line until line 2146 (not included), as the rest of the matches found by grep belong to more information after the end of the book.

```
linesQ <- lines[21:2145]
```

We can take a quick look at the first and last lines to make sure we've made the correct subset.

```
length(linesQ)
linesQ[1:10]
linesQ[2116:2125]
```

```
> length(linesQ)
[1] 2125
> linesQ[1:10]
 [1] ""                                                                 ""
 [3] ""                                                                 ""
 [5] "Produced by an anonymous Project Gutenberg volunteer." ""
 [7] ""                                                                 ""
 [9] ""                                                                 ""
> linesQ[2116:2125]
 [1] "aquí adelante me sucediere avisaré a vuestra merced.}"
 [2] ""
 [3] ""
 [4] ""
 [5] ""
 [6] ""
 [7] ""
 [8] "End of the Project Gutenberg EBook of La vida de Lazarillo de tormes y de"
 [9] "sus fortunas y adversidades, by Unknown"
[10] ""
```

By looking at this we realize we can actually trim both the start and the end a bit further, so more lines of irrelevant text will be cut. The book also features a Prologue we are by no means interested in at the moment (though, just for the sake of seeing if there is any difference between the author's preliminary speech and his actual writing, we will come back to it later), which we also do not want around. Fortunately, each of the seven chapters found in the book goes by the name of "Tratado <number name>", so we can try to find the start of each of these chapters by greping that as well.

```
> grep(pattern = "Tratado", linesQ, fixed = TRUE)
[1]   78  575  981 1659 1677 1993 2021
```

We get exactly seven matches: one for each of our chapters. Let's move on. We will use this knowledge to trim the start of our text; we already know how many rows we have to cut off the end.

```
linesQ <- linesQ[-c(1:77)]
linesQ <- linesQ[-c(2040:2048)]
length(linesQ)
linesQ[1:10]
linesQ[2030:2039]
```

```
> length(linesQ)
[1] 2039
> linesQ[1:10]
 [1] "Tratado Primero"
 [2] "Cuenta Lázaro su vida, y cuyo hijo fue"
 [3] ""
 [4] ""
 [5] ""
 [6] "Pues sepa V.M. ante todas cosas que a mí llaman Lázaro de Tormes, hijo"
 [7] "de Tomé González y de Antona Pérez, naturales de Tejares, aldea de"
 [8] "Salamanca. Mi nacimiento fue dentro del río Tormes, por la cual causa"
 [9] "tomé el sobrenombre, y fue desta manera. Mi padre, que Dios perdone,"
[10] "tenía cargo de proveer una molienda de una aceña, que está ribera de"
> linesQ[2030:2039]
 [1] "mujer como vive dentro de las puertas de Toledo. Quien otra cosa me"
 [2] "dijere, yo me mataré con él.\""
 [3] ""
 [4] "Desta manera no me dicen nada, y yo tengo paz en mi casa."
 [5] ""
 [6] "Esto fue el mesmo año que nuestro victorioso Emperador en esta insigne"
 [7] "ciudad de Toledo entró y tuvo en ella cortes, y se hicieron grandes"
 [8] "regocijos, como vuestra merced habrá oído. Pues en este tiempo estaba"
 [9] "en mi prosperidad y en la cumbre de toda buena fortuna{, de lo que de"
[10] "aquí adelante me sucediere avisaré a vuestra merced.}"
```

The source already specifies that our file is in UTF-8, but we will perform some validation just to be safe:

```
> linesQ[!utf8_valid((linesQ))]
character(0)
> linesQ_NFC <- utf8_normalize(linesQ)
> sum(linesQ_NFC != linesQ)
[1] 0
```

And we remove spaces and tabs:

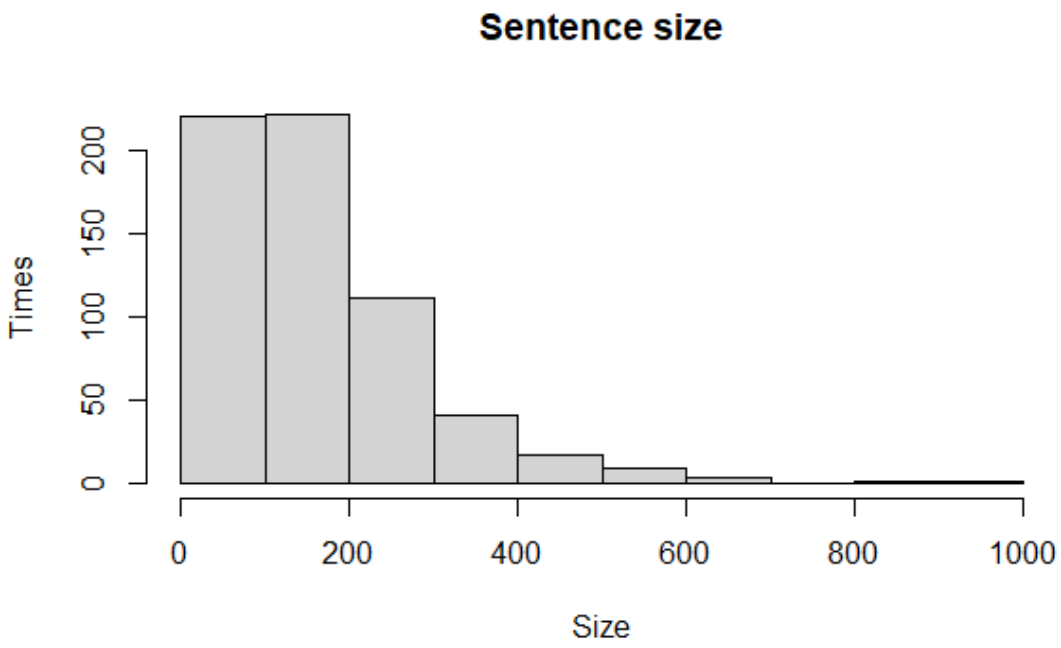
```
substring(paras[1], 1, 50)
parclean <- gsub("[\n]{1,}"," ", paras)
paras <- gsub("[\n]{2,}"," ", parclean)
substring(paras[1], 1, 50)
```

```
> substring(paras[1], 1, 50)
[1] "Tratado Primero\nCuenta Lázaro su vida, y cuyo hijo"
> parclean <- gsub("[\n]{1,}"," ", paras)
> paras <- gsub("[\n]{2,}"," ", parclean)
> substring(paras[1], 1, 50)
[1] "Tratado Primero Cuenta Lázaro su vida, y cuyo hijo"
```

2. Sentence analysis

After initializing spacy with our Python environment and tokenizing our text into sentences, we get a total of 14 texts. While it is true that it is a very short book, it is quite a small number of divisions to make. We will take a look at the size of these “sentences”:

```
sentences <- spacy_tokenize(paras, what="sentence")
v_sentences <- unlist(sentences)
nsentences <- length(v_sentences)
sum(v_sentences=="")

histSentences <- hist(nchar(v_sentences),
     main = "Sentence size",
     xlab = "size",
     ylab = "Times"
     )
```



3. Token analysis

These sentences are too large, we are interested in breaking the text down further. For that we will use regular tokens, words.

```
n_tokens <- spacy_tokenize(paras)
v_tokens <- unlist(n_tokens) #22.859 tokens
v_tokens[1:10]
```

```
> v_tokens[1:10]
   text11    text12    text13    text14    text15    text16    text17
"Tratado" "Primero"  "Cuenta"  "Lázaro"      "su"    "vida"       ","
   text18    text19   text110
      "y"    "cuyo"    "hijo"
```

Though we have a big number of words in ours short tale, the number of unique tokens pales in comparison:

```
> length(v_tokens)
[1] 22859
> length(unique(v_tokens))
[1] 3898
```

It is interesting to consider what the real number would be, without punctuation of any kind:

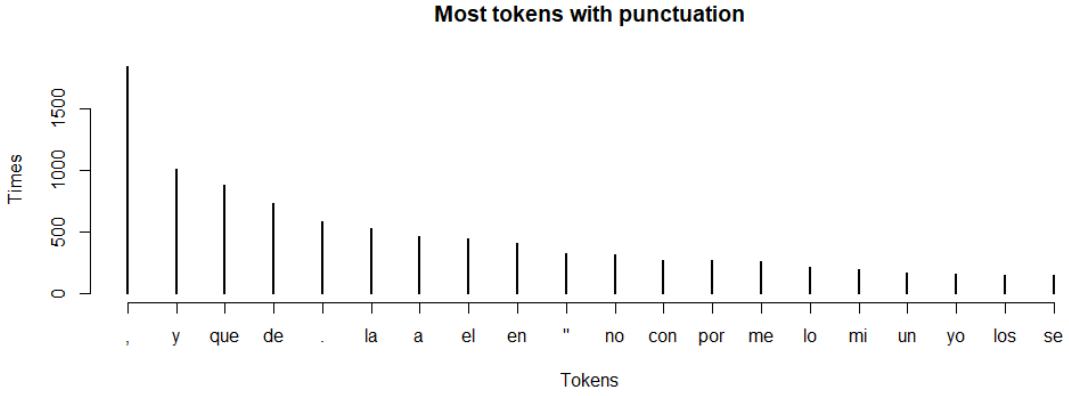
```
tokens_no_punct <- spacy_tokenize(paras, remove_punct = TRUE)
v_tokens_no_punct <- unlist(tokens_no_punct)
length(v_tokens_no_punct)
length(unique(v_tokens_no_punct))
```

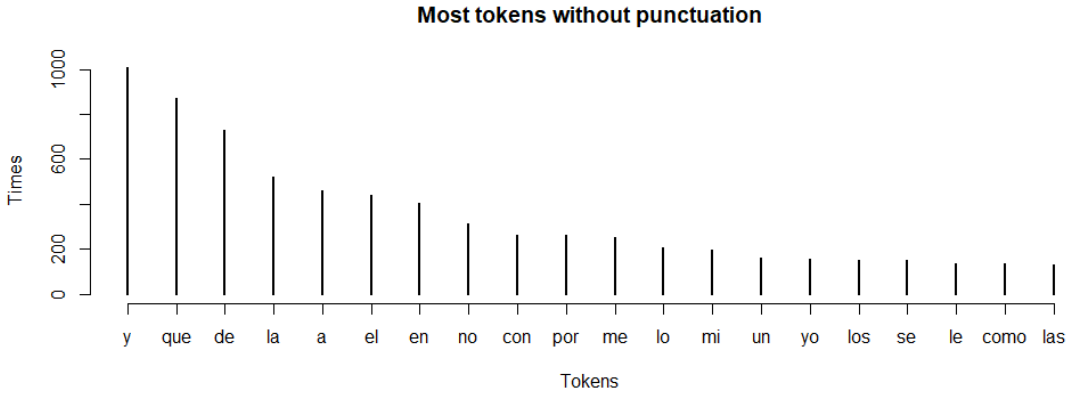
```
> length(v_tokens_no_punct)
[1] 19621
> length(unique(v_tokens_no_punct))
[1] 3882
```

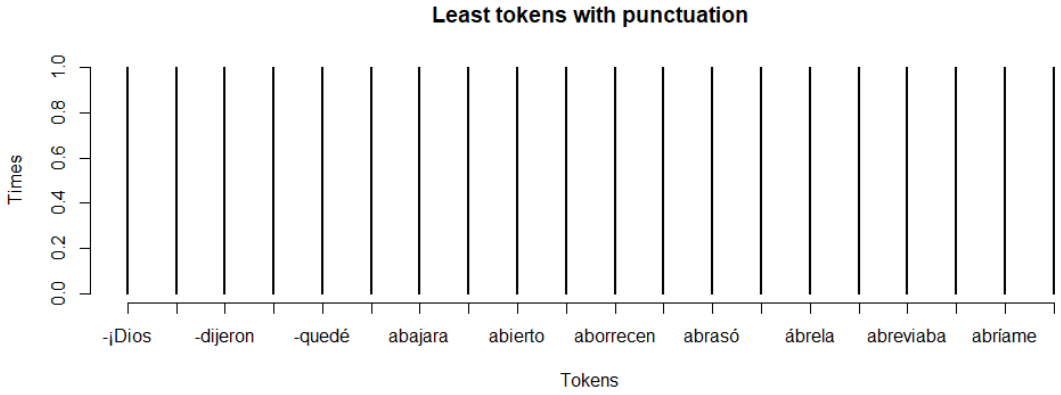
The difference in unique characters in both sets of tokens is of only 16, which tells us that that is the number of different punctuation characters we can find in the text. The 3238 tokens we have removed through this method equal to about 14.16% of the original token set, or about 1 every 7 tokens, an important slice.

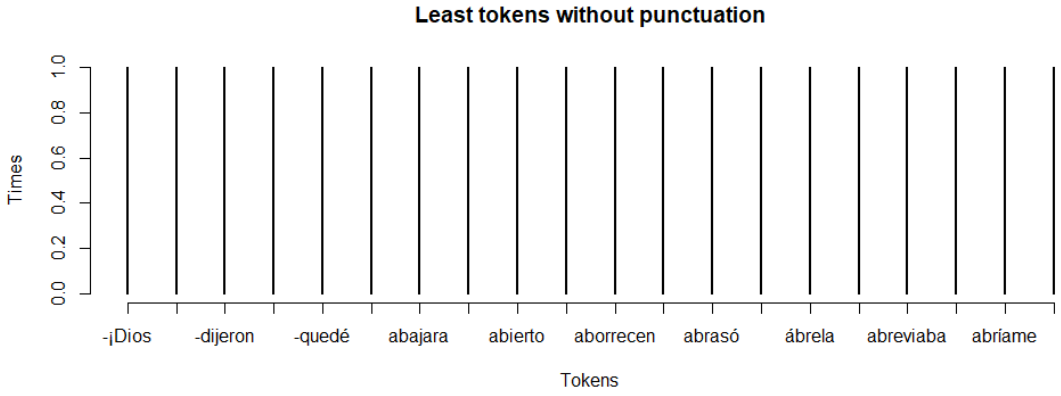
We can plot the tokens we have in both subsets to see their variance. We will take a look at the most and least frequent:

```
tableTokens <- head(sort(table(v_tokens), decreasing = TRUE), n = 20)
tableNPTokens <- head(sort(table(v_tokens_no_punct), decreasing = TRUE), n = 20)
tableLeastTokens <- head(sort(table(v_tokens), decreasing = FALSE), n = 20)
tableLeastNPTokens <- head(sort(table(v_tokens_no_punct), decreasing = FALSE), n = 20)
```









There is a big overlap between both pairs of plots. First of all, we can see that "," and "." and even " " " itself are very common in our text, which makes sense due to its nature. The version without punctuation replaces these symbols with the next most frequent terms in the list, "le", "como" and "las". Since it is a Spanish text, we can also corroborate that "los" should be higher than "las" in the list, as plurals tend to be in masculine form even when they involve different genders, and it actually is four places higher in the ranking. Then, we take a look at the least frequent terms. Obviously, in order to even make it into the token list, there has to be a single occurrence of that term, which is the case of all these words. We can also realize that some of the plotted words, like "¡Dios" and "-dijeron" have punctuation in them, which the tokenizer didn't recognize as such, but as part of a normal word.

4. Chapter by chapter analysis

So, we have taken a look at the tokens in the text as a whole, but we are more interested in how that distribution changes along the chapters. Let's remember that the book has a total of seven of these, easily identifiable by the preface "Tratado ___":

```
> grep(pattern = "Tratado ", linesQ, fixed = TRUE)
[1]    1  498  904 1582 1600 1916 1944
> linesQ[1:20]
 [1] "Tratado Primero"
 [2] "Cuenta Lázaro su vida, y cuyo hijo fue"
 [3] ""
 [4] ""
 [5] ""
 [6] "Pues sepa V.M. ante todas cosas que a mí llaman Lázaro de Tormes, hijo"
 [7] "de Tomé González y de Antona Pérez, naturales de Tejares, aldea de"
 [8] "Salamanca. Mi nacimiento fue dentro del río Tormes, por la cual causa"
 [9] "tomé el sobrenombre, y fue desta manera. Mi padre, que Dios perdone,"
[10] "tenía cargo de proveer una molienda de una aceña, que está ribera de"
[11] "aquel río, en la cual fue molinero más de quince años; y estando mi"
[12] "madre una noche en la aceña, preñada de mí, tomóle el parto y parióme"
[13] "allí: de manera que con verdad puedo decir nacido en el río. Pues"
[14] "siendo yo niño de ocho años, achacaron a mi padre ciertas sangrías mal"
[15] "hechas en los costales de los que allí a moler venían, por lo que fue"
[16] "preso, y confesó y no negó y padeció persecución por justicia. Espero"
[17] "en Dios que está en la Gloria, pues el Evangelio los llama"
[18] "bienaventurados. En este tiempo se hizo cierta armada contra moros,"
[19] "entre los cuales fue mi padre, que a la sazón estaba desterrado por el"
[20] "desastre ya dicho, con cargo de acemilero de un caballero que allá fue,"
```

Let's make a list with all the chapters by using this information.

```
cap1 <- c(linesQ[1:497])
cap2 <- c(linesQ[498:903])
cap3 <- c(linesQ[904:1581])
cap4 <- c(linesQ[1582:1599])
cap5 <- c(linesQ[1600:1915])
cap6 <- c(linesQ[1916:1944])
cap7 <- c(linesQ[1944:length(linesQ)])

caps <- list()
caps[[1]] <- c(cap1)
caps[[2]] <- c(cap2)
caps[[3]] <- c(cap3)
caps[[4]] <- c(cap4)
caps[[5]] <- c(cap5)
caps[[6]] <- c(cap6)
caps[[7]] <- c(cap7)
```

These chapters are quite uneven in size; however, the plot of each chapter is different from the rest, which should in theory keep some of the vocabulary in each chapter relatively self-contained. We could test this theory by trying both ideas: splitting the book into regular, chapter based halves (chapters 1-3 vs chapters 4-7, making it 1581 vs 458 lines) and splitting it in two more "fair" halves (chapters 1 and 2 vs chapters 3-7; 903 vs 1136 lines).

Chapter based halves

```
caps1st <- paste(unlist(caps[1:3]))
caps2nd <- paste(unlist(caps[4:7]))

model <- bpe(unlist(caps[1:3]))
subtoks2 <- bpe_encode(model, x = caps2nd, type = "subwords")
head(unlist(subtoks2), n = 50)
```

```
> head(unlist(subtoks2), n = 50)
 [1] "▁Tratado" "▁Cu"      "ar"       "to"       "▁Cómo"    "▁Lázaro"  "▁se"      "▁asentó"  "▁con"
[10] "▁un"      "▁f"       "ra"       "i"        "le"       "▁de"      "▁la"      "▁M"       "er"
[19] "ced"      ","        "▁y"       "▁de"      "▁lo"      "▁que"     "▁le"      "▁acaeció" "▁con"
[28] "▁él"      "▁H"       "u"        "b"        "e"        "▁de"      "▁buscar"  "▁el"      "▁cu"
[37] "ar"       "to,"      "▁y"       "▁éste"    "▁fue"     "▁un"      "▁f"       "ra"       "i"
[46] "le"       "▁de"      "▁la"      "▁M"       "er"
```

Fair halves

```
caps1stfair <- paste(unlist(caps[1:2]))
caps2ndfair <- paste(unlist(caps[3:7]))
model <- bpe(unlist(caps[1:2]))
subtoksfair <- bpe_encode(model, x = caps2ndfair, type = "subwords")
head(unlist(subtoksfair), n = 50)
```

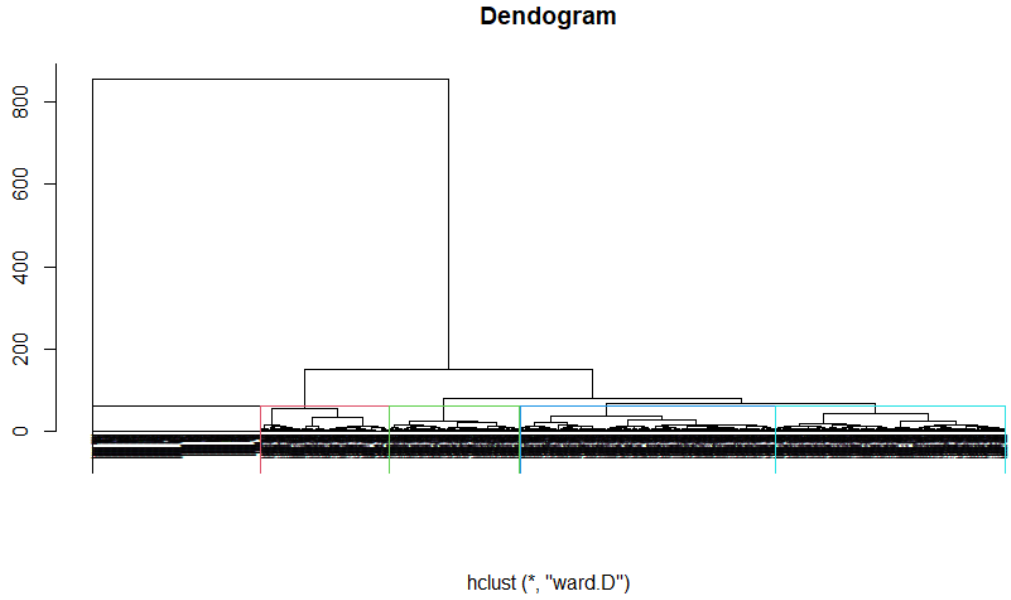
```
> head(unlist(subtoksfair), n = 50)
 [1] "▁Tratado"  "▁T"        "er"        "cer"       "o"         "▁Cómo"     "▁Lázaro"   "▁se"
 [9] "▁asentó"   "▁con"      "▁un"       "▁esc"      "ud"        "ero,"      "▁y"        "▁de"
[17] "▁lo"       "▁que"      "▁le"       "▁acaeció"  "▁con"      "▁él"       "▁De"       "sta"
[25] "▁manera"   "▁me"       "▁fue"      "▁for"      "z"         "ado"       "▁sacar"    "▁fuer"
[33] "zas"       "▁de"       "▁flaqueza" "▁y,"       "▁poco"     "▁a"        "▁poco,"    "▁con"
[41] "▁ayuda"    "▁de"       "▁las"      "▁buenas"   "▁g"        "ent"       "es"        "▁di"
[49] "▁comigo"   "▁en"
```

Feeding our lines directly to the model instead of chapters gives us the same result. Since our text does not have an immense size, we can make our corpus line by line, and study how the author's writing changes:

```
texts_lines <- unlist(linesQ)
names(texts_lines) <- paste("Línea ", 1:length(texts_lines))
corpus_lines <- corpus(texts_lines)
docvars(corpus_lines, field = "Línea") <- 1:length(texts_lines)
corpus_lines
```

```
> corpus_lines
Corpus consisting of 2,039 documents and 1 docvar.
Línea  1 :
"Tratado Primero"

Línea  2 :
"Cuenta Lázaro su vida, y cuyo hijo fue"

Línea  3 :
""

Línea  4 :
""

Línea  5 :
""

Línea  6 :
"Pues sepa V.M. ante todas cosas que a mí llaman Lázaro de To..."

[ reached max_ndoc ... 2,033 more documents ]
```

```
texts_lines <- unlist(linesQ)
names(texts_lines) <- paste("Línea ", 1:length(texts_lines))
corpus_lines <- corpus(texts_lines)
docvars(corpus_lines, field = "Línea") <- 1:length(texts_lines)
corpus_lines

dfm_lines <- dfm(tokens(corpus_lines),)
distMatrixLines <- dist(as.matrix(dfm_lines))
groups <- hclust(distMatrixLines, method="ward.D")

plot(groups,
     cex = 0.25,
     hang = -1,
     xlab = "",
     ylab = "",
     main = "")
rect.hclust(groups, k = 5)
```



5. Feature frequency

If we take a look at the frequency, we will still find that commas are at the top, along with several connectors. We will remove them all from the list to get more realistic results.

```
topfeatures(dfm_lines)
dfm_lines_NP <- dfm(tokens(corpus_lines, remove_punct = TRUE), )
dfm_lines_NP2 <- dfm_remove(dfm_lines_NP, stopwords("es"))
topfeatures(dfm_lines_NP2)
topfeatures(dfm_lines_NP2, decreasing = FALSE)
```

```
> topfeatures(dfm_lines)
   ,    y  que   de    .   la    a   el   en   no
1837 1118  876  742  593  525  475  452  423  328
> topfeatures(dfm_lines_NP2)
  mas  bien   amo  dios    si señor  casa  pues   tan   ser
   88    77    77    76    74    66    55    54    53    52
> topfeatures(dfm_lines_NP2, decreasing = FALSE)
      cuyo    gonzález      antona       pérez     tejares       aldea  nacimiento sobrenombre    molienda      ribera
         1           1           1           1           1           1           1           1           1           1
```

We can split our corpus in two (following our fair halves idea):

```
corpus_pt1 <- corpus_subset(corpus_lines,
                            Línea < 904) #Chapters 1-2
corpus_pt2 <- corpus_subset(corpus_lines,
                            Línea > 903) #Chapters 3-7

# First half vs second half
dfm_pt1NP <- dfm(tokens(corpus_pt1, remove_punct = TRUE))
dfm_pt2NP <- dfm(tokens(corpus_pt2, remove_punct = TRUE))

dfm_pt1NP <- dfm_remove(dfm_pt1NP, stopwords("es"))
dfm_pt2NP <- dfm_remove(dfm_pt2NP, stopwords("es"))

topfeatures(dfm_pt1NP)
topfeatures(dfm_pt2NP)
topfeatures(dfm_pt1NP, decreasing = FALSE)
topfeatures(dfm_pt2NP, decreasing = FALSE)
```

```
> topfeatures(dfm_pt1NP)
  mas ciego    si  dios   ser   tan   día   tal  vino  pues
   40    33    33    30    28    27    26    25    24    23
> topfeatures(dfm_pt2NP)
  amo  bien señor   mas  dios    si  casa  pues  ansí   día
   57    55    55    48    46    41    37    31    27    26
> topfeatures(dfm_pt1NP, decreasing = FALSE)
      cuyo    gonzález      antona       pérez   naturales     tejares       aldea  nacimiento
         1           1           1           1           1           1           1           1
sobrenombre     proveer
         1           1
> topfeatures(dfm_pt2NP, decreasing = FALSE)
  forzado   fuerzas  flaqueza     dende     cerró    herida   bellaco gallofero    sirvas
        1         1         1         1         1         1         1         1         1
  hallará
        1
```

Finding no match in the bottom features (which is to be expected, as several terms only appear once) and “Dios”, ““si”, día” and “pues” as top features in both halves. Now, we will test whether this is the same in each chapter:

```
corpus_ch1 <- corpus_subset(corpus_lines,
                            Línea < 498) #Chapter 1
corpus_ch2 <- corpus_subset(corpus_lines,
                            ((497 < Línea) & (Línea < 904))) #Chapter 2
corpus_ch3 <- corpus_subset(corpus_lines,
                            ((903 < Línea) & (Línea < 1581))) #Chapter 3
corpus_ch4 <- corpus_subset(corpus_lines,
                            ((1580 < Línea) & (Línea < 1600))) #Chapter 4
corpus_ch5 <- corpus_subset(corpus_lines,
                            ((1599 < Línea) & (Línea < 1916))) #Chapter 5
corpus_ch6 <- corpus_subset(corpus_lines,
                            ((1915 < Línea) & (Línea < 1944))) #Chapter 6
corpus_ch7 <- corpus_subset(corpus_lines,
                            1943 < Línea) #Chapter 7

dfm_ch1NP <- dfm(tokens(corpus_ch1, remove_punct = TRUE))
dfm_ch2NP <- dfm(tokens(corpus_ch2, remove_punct = TRUE))
dfm_ch3NP <- dfm(tokens(corpus_ch3, remove_punct = TRUE))
dfm_ch4NP <- dfm(tokens(corpus_ch4, remove_punct = TRUE))
dfm_ch5NP <- dfm(tokens(corpus_ch5, remove_punct = TRUE))
dfm_ch6NP <- dfm(tokens(corpus_ch6, remove_punct = TRUE))
dfm_ch7NP <- dfm(tokens(corpus_ch7, remove_punct = TRUE))

dfm_ch1NP <- dfm_remove(dfm_ch1NP, stopwords("es"))
dfm_ch2NP <- dfm_remove(dfm_ch2NP, stopwords("es"))
dfm_ch3NP <- dfm_remove(dfm_ch3NP, stopwords("es"))
dfm_ch4NP <- dfm_remove(dfm_ch4NP, stopwords("es"))
dfm_ch5NP <- dfm_remove(dfm_ch5NP, stopwords("es"))
dfm_ch6NP <- dfm_remove(dfm_ch6NP, stopwords("es"))
dfm_ch7NP <- dfm_remove(dfm_ch7NP, stopwords("es"))

topfeatures(dfm_ch1NP)
topfeatures(dfm_ch2NP)
topfeatures(dfm_ch3NP)
topfeatures(dfm_ch4NP)
topfeatures(dfm_ch5NP)
topfeatures(dfm_ch6NP)
topfeatures(dfm_ch7NP)
```

```
> topfeatures(dfm_ch1NP)
ciego   mas  vino   mal    si   tal  dios   ser   tan  pues
   28    21    20    15    15    14    13    13    13    12
> topfeatures(dfm_ch2NP)
  mas    si  dios   día  arca   ser   tan noche llave   amo
   19    18    17    16    16    15    14    14    13    13
> topfeatures(dfm_ch3NP)
  dios   bien    mas   casa  señor    amo     si   pues    ser aunque
    34     34     33     32     30     29     25     20     19     17
> topfeatures(dfm_ch4NP)
  cuarto   fraile   merced     éste     digo convento  zapatos  tratado     cómo   lázaro
       2        2        2        2        2        2        2        1        1        1
> topfeatures(dfm_ch5NP)
     amo    señor     bula     ansí alguacil       si     bien     cruz    tomar    hacer
      21       19       17       14       14       13       13       12       11       11
> topfeatures(dfm_ch6NP)
 capellán      buen       día      asno    cuatro       amo   treinta maravedís      bien    oficio
        2         2         2         2         2         2         2         2         2         2
> topfeatures(dfm_ch7NP)
  bien  señor  mujer    amo   dios    día    mas oficio merced  buena
     6      6      6      5      5      5      4      4      4      4
```

As we guessed, each chapter doesn't really overlap with the others in terms of vocabulary, except for words like, yet again, "Dios", "si", "amo" and "señor". Given the plot, not very surprising. But what about the book versus the discarded prologue?

```
grep(pattern = "Prólogo", lines, fixed = TRUE)
grep(pattern = "Tratado", lines, fixed = TRUE)
linesR <- lines[52:97]
linesR
```

```
> linesR
 [1] "Prólogo"
 [2] ""
 [3] ""
 [4] ""
 [5] "Yo por bien tengo que cosas tan señaladas, y por ventura nunca oídas ni"
 [6] "vistas, vengan a noticia de muchos y no se entierren en la sepultura"
 [7] "del olvido, pues podría ser que alguno que las lea halle algo que le"
 [8] "agrade, y a los que no ahondaren tanto los deleite; y a este propósito"
 [9] "dice Plinio que no hay libro, por malo que sea, que no tenga alguna"
[10] "cosa buena; mayormente que los gustos no son todos unos, mas lo que uno"
[11] "no come, otro se pierde por ello. Y así vemos cosas tenidas en poco de"
[12] "algunos, que de otros no lo son. Y esto, para ninguna cosa se debría"
[13] "romper ni echar a mal, si muy detestable no fuese, sino que a todos se"
```

```
texts_lines_prolog <- unlist(linesR)
names(texts_lines_prolog) <- paste("Línea ", 1:length(texts_lines_prolog))
corpus_lines_prolog <- corpus(texts_lines_prolog)
docvars(corpus_lines_prolog, field = "Línea") <- 1:length(texts_lines_prolog)
corpus_lines_prolog
```

```
> corpus_lines_prolog
Corpus consisting of 46 documents and 1 docvar.
Línea  1 :
"Prólogo"

Línea  2 :
""

Línea  3 :
""

Línea  4 :
""

Línea  5 :
"Yo por bien tengo que cosas tan señaladas, y por ventura nun..."

Línea  6 :
"vistas, vengan a noticia de muchos y no se entierren en la s..."

[ reached max_ndoc ... 40 more documents ]
```

```
dfm_prolog_NP <- dfm(tokens(corpus_lines_prolog, remove_punct = TRUE))
dfm_prolog_NP <- dfm_remove(dfm_prolog_NP, stopwords("es"))
topfeatures(dfm_prolog_NP)
topfeatures(dfm_prolog_NP, decreasing = FALSE)
```

```
> topfeatures(dfm_prolog_NP)
      si      pues       mas       ser       así      bien     cosas   noticia propósito      dice
       6         4         4         3         3         2         2         2         2         2
> topfeatures(dfm_prolog_NP, decreasing = FALSE)
  prólogo       tan señaladas   ventura     nunca     oídas    vistas    vengan entierren sepultura
        1         1         1         1         1         1         1         1         1         1
```

Even though it is a small sample, it is close in size to chapters 4, 6 and 7:

```
> topfeatures(dfm_ch4NP)
  cuarto   fraile   merced     éste     digo convento  zapatos  tratado     cómo   lázaro 
       2        2        2        2        2        2        2        1        1        1 
> topfeatures(dfm_ch6NP)
 capellán      buen       día      asno    cuatro       amo   treinta maravedís      bien    oficio 
       2         2         2         2         2         2         2         2         2         2 
> topfeatures(dfm_ch7NP)
  bien  señor  mujer    amo   dios    día    mas oficio merced  buena 
     6      6      6      5      5      5      4      4      4      4 
```

And we observe that in these four sections, we find variations of “bien” at the top: “bien”, “buena” and “buen”, as well as a small overlap between the prologue and chapter 7 with “mas” and “digo” and “dice” in the prologue and chapter 4.

We can conclude that once we remove stopwords, punctuation, and connectors of every kind in our language, what is left behind in this particular case is God, ironically. We also find names used to refer to people at the time, such as “señor”, “amo” and “merced”.

6. Session info and code

```
> sessionInfo()
R version 4.0.2 (2020-06-22)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 10 x64 (build 19042)

Matrix products: default

locale:
[1] LC_COLLATE=Spanish_Spain.1252  LC_CTYPE=Spanish_Spain.1252    LC_MONETARY=Spanish_Spain.1252
[4] LC_NUMERIC=C                   LC_TIME=Spanish_Spain.1252

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] quanteda_3.2.0       tokenizers.bpe_0.1.0 udpipe_0.8.8         spacyr_1.2.1
[5] utf8_1.1.4

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.8          here_1.0.1          lattice_0.20-41     png_0.1-7           rprojroot_2.0.2
 [6] rappdirs_0.3.3      grid_4.0.2          jsonlite_1.7.1      magrittr_1.5        RcppParallel_5.1.5
[11] stringi_1.5.3       data.table_1.13.0   Matrix_1.2-18       reticulate_1.23     stopwords_2.3
[16] fastmatch_1.1-3     tools_4.0.2         tinytex_0.26        xfun_0.18           compiler_4.0.2
```

```
#install.packages("lessR")

#install.packages("spacyr")

#install.packages("quanteda")

library(utf8)

library(spacyr)

library(tokenizers.bpe)

library(quanteda)


# We input our URL. Originally, the URL linked in the email led to a file

# encoded in ISO-8859-1. We could format it, but since we have an UTF-8

# mirror at our disposal on the same page, we will use that instead.


urlLazarillo <- "https://www.gutenberg.org/cache/epub/320/pg320.txt"

lines <- readLines(urlLazarillo, encoding = "UTF-8")

grep(pattern = "***", lines, fixed = TRUE)

linesQ <- lines[21:2145]

length(linesQ)

linesQ[1:10]

linesQ[2116:2125]


# We skip the prologue and preliminary notes by greping the

# first words of the first chapter. "Tratado"


grep(pattern = "Tratado", linesQ, fixed = TRUE) # [1]   78  575  981 1659 1677 1993 2021

linesQ <- linesQ[-c(1:77)]

linesQ <- linesQ[-c(2040:2048)]

length(linesQ)
```

```
linesQ[78:88]
linesQ[2030:2039]

paste(linesQ[1:5], collapse = " ")

linesQ[!utf8_valid((linesQ))]
linesQ_NFC <- utf8_normalize(linesQ)
sum(linesQ_NFC != linesQ)

stringQ <- paste(linesQ, collapse ="\n")
paras <- unlist(strsplit(stringQ, "\n\n\n", fixed = TRUE))
parEmpty <- which(paras == "")
length(paras)

substring(paras[1], 1, 50)
parclean <- gsub("[\n]{1,}"," ", paras)
paras <- gsub("[\n]{2,}"," ", parclean)
substring(paras[1], 1, 50)

spacy_install()
spacy_download_langmodel('es')
spacy_initialize(model = "es_core_news_sm")

sentences <- spacy_tokenize(paras, what="sentence")
v_sentences <- unlist(sentences)
nsentences <- length(v_sentences) #626
sum(v_sentences=="") #1

#v_sentences <- v_sentences[-which(v_sentences=="")]
histSentences <- hist(nchar(v_sentences),
     main = "Sentence size",
     xlab = "Size",
     ylab = "Times"
     )

# Number of tokens

n_tokens <- spacy_tokenize(paras)
v_tokens <- unlist(n_tokens)
```

```
v_tokens[1:10]
length(v_tokens)
length(unique(v_tokens))
tokens_no_punct <- spacy_tokenize(paras, remove_punct = TRUE)
v_tokens_no_punct <- unlist(tokens_no_punct)
length(v_tokens_no_punct)
length(unique(v_tokens_no_punct))

tableTokens <- head(sort(table(v_tokens), decreasing = TRUE), n = 20)
tableNPTokens <- head(sort(table(v_tokens_no_punct), decreasing = TRUE), n = 20)
tableLeastTokens <- head(sort(table(v_tokens), decreasing = FALSE), n = 20)
tableLeastNPTokens <- head(sort(table(v_tokens_no_punct), decreasing = FALSE), n = 20)

# Token plots

plot(tableTokens,
     xlab = "Tokens",
     ylab = "Times",
     main = "Most tokens with punctuation")

plot(tableNPTokens,
     xlab = "Tokens",
     ylab = "Times",
     main = "Most tokens without punctuation")

plot(tableLeastTokens,
     xlab = "Tokens",
     ylab = "Times",
     main = "Least tokens with punctuation")

plot(tableLeastNPTokens,
     xlab = "Tokens",
     ylab = "Times",
     main = "Least tokens without punctuation")

spacy_finalize()

# Chapter analysis
```

```
grep(pattern = "Tratado ", linesQ, fixed = TRUE)
linesQ[1:20]

cap1 <- c(linesQ[1:497])
cap2 <- c(linesQ[498:903])
cap3 <- c(linesQ[904:1581])
cap4 <- c(linesQ[1582:1599])
cap5 <- c(linesQ[1600:1915])
cap6 <- c(linesQ[1916:1944])
cap7 <- c(linesQ[1944:length(linesQ)])

caps <- list()
caps[[1]] <- c(cap1)
caps[[2]] <- c(cap2)
caps[[3]] <- c(cap3)
caps[[4]] <- c(cap4)
caps[[5]] <- c(cap5)
caps[[6]] <- c(cap6)
caps[[7]] <- c(cap7)

# Regular halves

caps1st <- paste(unlist(caps[1:3]))
caps2nd <- paste(unlist(caps[4:7]))

model <- bpe(unlist(caps[1:3]))
subtoks2 <- bpe_encode(model, x = caps2nd, type = "subwords")
head(unlist(subtoks2), n = 50)

# Fair halves

caps1stfair <- paste(unlist(caps[1:2]))
caps2ndfair <- paste(unlist(caps[3:7]))

model <- bpe(unlist(caps[1:2]))
subtoksfair <- bpe_encode(model, x = caps2ndfair, type = "subwords")
head(unlist(subtoksfair), n = 50)

model2 <- bpe(unlist(linesQ))
```

```
subtoks3 <- bpe_encode(model2, x = caps2ndfair, type = "subwords")
head(unlist(subtoks3), n = 50)

# Corpus definition

texts_lines <- unlist(linesQ)
names(texts_lines) <- paste("Línea ", 1:length(texts_lines))
corpus_lines <- corpus(texts_lines)
docvars(corpus_lines, field = "Línea") <- 1:length(texts_lines)
corpus_lines

# Dendogram setup

dfm_lines <- dfm(tokens(corpus_lines),)
distMatrixLines <- dist(as.matrix(dfm_lines))
groups <- hclust(distMatrixLines, method="ward.D")

plot(groups,
     cex = 0.25,
     hang = -1,
     xlab = "",
     ylab = "",
     main = "Dendogram")
rect.hclust(groups, k = 5, border = 1:5)

topfeatures(dfm_lines)
dfm_lines_NP <- dfm(tokens(corpus_lines, remove_punct = TRUE), )
dfm_lines_NP2 <- dfm_remove(dfm_lines_NP, stopwords("es"))
topfeatures(dfm_lines_NP2)
topfeatures(dfm_lines_NP2, decreasing = FALSE)

# Using docvars with fair halves
corpus_pt1 <- corpus_subset(corpus_lines,
                            Línea < 904) #Chapters 1-2
corpus_pt2 <- corpus_subset(corpus_lines,
                            Línea > 903) #Chapters 3-7

# First half vs second half
dfm_pt1NP <- dfm(tokens(corpus_pt1, remove_punct = TRUE))
```

```
dfm_pt2NP <- dfm(tokens(corpus_pt2, remove_punct = TRUE))

dfm_pt1NP <- dfm_remove(dfm_pt1NP, stopwords("es"))
dfm_pt2NP <- dfm_remove(dfm_pt2NP, stopwords("es"))

topfeatures(dfm_pt1NP)
topfeatures(dfm_pt2NP)
topfeatures(dfm_pt1NP, decreasing = FALSE)
topfeatures(dfm_pt2NP, decreasing = FALSE)

# Chapter vs chapter
corpus_ch1 <- corpus_subset(corpus_lines,
                            Línea < 498) #Chapter 1
corpus_ch2 <- corpus_subset(corpus_lines,
                            ((497 < Línea) & (Línea < 904))) #Chapter 2
corpus_ch3 <- corpus_subset(corpus_lines,
                            ((903 < Línea) & (Línea < 1581))) #Chapter 3
corpus_ch4 <- corpus_subset(corpus_lines,
                            ((1580 < Línea) & (Línea < 1600))) #Chapter 4
corpus_ch5 <- corpus_subset(corpus_lines,
                            ((1599 < Línea) & (Línea < 1916))) #Chapter 5
corpus_ch6 <- corpus_subset(corpus_lines,
                            ((1915 < Línea) & (Línea < 1944))) #Chapter 6
corpus_ch7 <- corpus_subset(corpus_lines,
                            1943 < Línea) #Chapter 7

dfm_ch1NP <- dfm(tokens(corpus_ch1, remove_punct = TRUE))
dfm_ch2NP <- dfm(tokens(corpus_ch2, remove_punct = TRUE))
dfm_ch3NP <- dfm(tokens(corpus_ch3, remove_punct = TRUE))
dfm_ch4NP <- dfm(tokens(corpus_ch4, remove_punct = TRUE))
dfm_ch5NP <- dfm(tokens(corpus_ch5, remove_punct = TRUE))
dfm_ch6NP <- dfm(tokens(corpus_ch6, remove_punct = TRUE))
dfm_ch7NP <- dfm(tokens(corpus_ch7, remove_punct = TRUE))

dfm_ch1NP <- dfm_remove(dfm_ch1NP, stopwords("es"))
dfm_ch2NP <- dfm_remove(dfm_ch2NP, stopwords("es"))
dfm_ch3NP <- dfm_remove(dfm_ch3NP, stopwords("es"))
dfm_ch4NP <- dfm_remove(dfm_ch4NP, stopwords("es"))
dfm_ch5NP <- dfm_remove(dfm_ch5NP, stopwords("es"))
```

```
dfm_ch6NP <- dfm_remove(dfm_ch6NP, stopwords("es"))
dfm_ch7NP <- dfm_remove(dfm_ch7NP, stopwords("es"))

topfeatures(dfm_ch1NP)
topfeatures(dfm_ch2NP)
topfeatures(dfm_ch3NP)
topfeatures(dfm_ch4NP)
topfeatures(dfm_ch5NP)
topfeatures(dfm_ch6NP)
topfeatures(dfm_ch7NP)

# Prologue vs book
grep(pattern = "Prólogo", lines, fixed = TRUE)
grep(pattern = "Tratado", lines, fixed = TRUE)
linesR <- lines[52:97]
linesR

texts_lines_prolog <- unlist(linesR)
names(texts_lines_prolog) <- paste("Línea ", 1:length(texts_lines_prolog))
corpus_lines_prolog <- corpus(texts_lines_prolog)
docvars(corpus_lines_prolog, field = "Línea") <- 1:length(texts_lines_prolog)
corpus_lines_prolog

dfm_prolog_NP <- dfm(tokens(corpus_lines_prolog, remove_punct = TRUE))
dfm_prolog_NP <- dfm_remove(dfm_prolog_NP, stopwords("es"))
topfeatures(dfm_prolog_NP)
topfeatures(dfm_prolog_NP, decreasing = FALSE)
```