NLP

El Lazarillo de Tormes

MARIA TERESA ROMO GALLEGO
30-1-2022

Link of the GitHub repo: https://github.com/tereromo/NLP_assignment

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

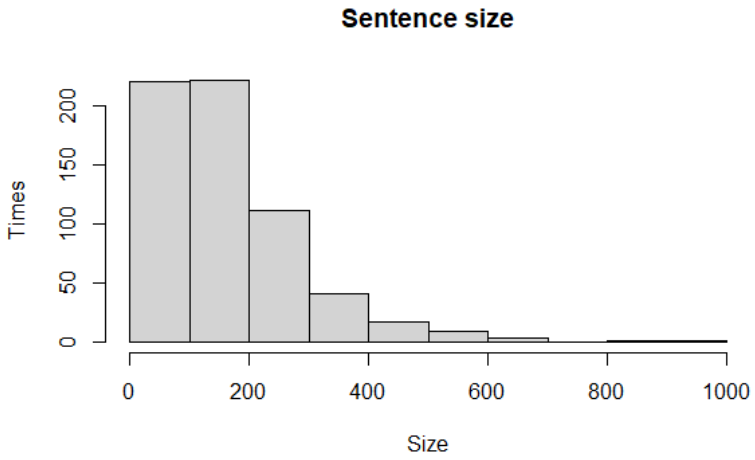
The book in question will be El Lazarillo de Tormes, as found in:

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. 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. 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.

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

After initializing spacy with our Python environment and tokenizing our text into sentences, we get a total of 14 texts.

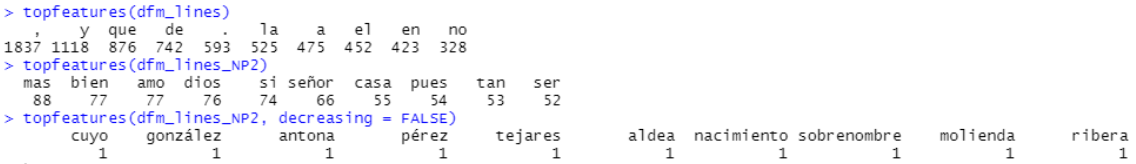


These sentences are too large, we are interested in breaking the text down further. For that we will use regular tokens, words. Though we have a big number of words in ours short tale, the number of unique tokens pales in comparison. It is interesting to consider what the real number would be, without punctuation of any kind. 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.

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.

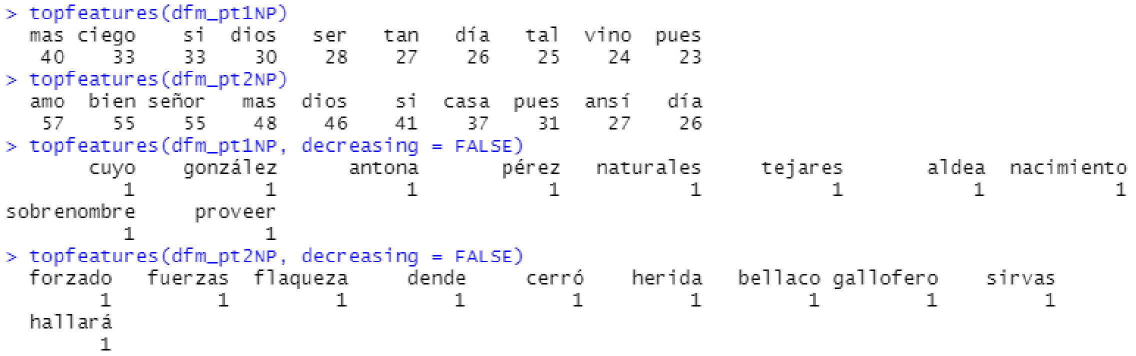
So, we have taken a look at the tokens in the text as a whole, but we are more interested in how that distribution changes along the chapters. Let's remember that the book has a total of seven of these, easily identifiable by the preface "Tratado ___". Let's make a list with all the chapters by using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

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.



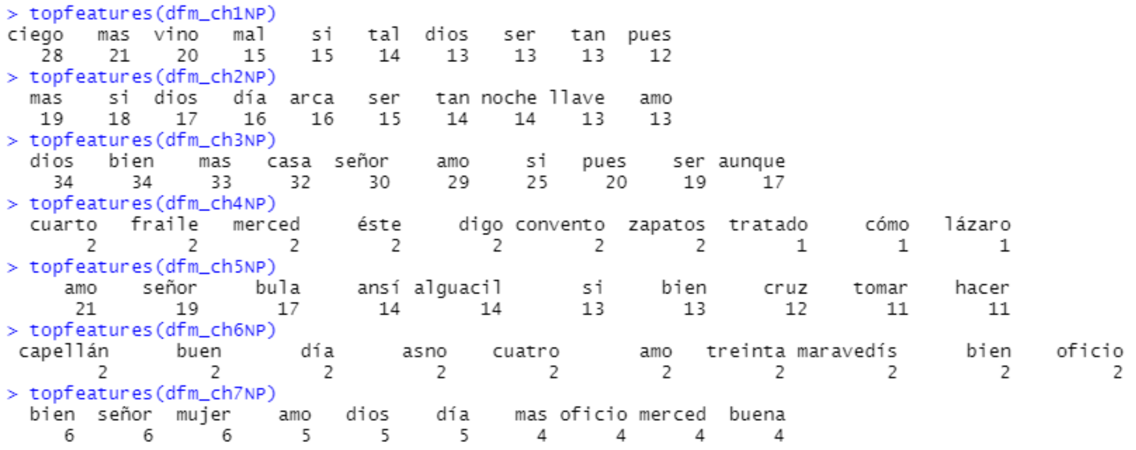
```
> topfeatures(dfm_lines)
   ,    y  que   de    .   la    a   el   en   no 
1837 1118  876  742  593  525  475  452  423  328 
> topfeatures(dfm_lines_NP2)
  mas  bien   amo  dios    si señor  casa  pues   tan   ser 
   88    77    77    76    74    66    55    54    53    52 
> topfeatures(dfm_lines_NP2, 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):



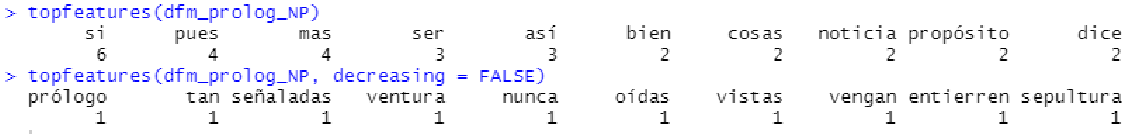
```
> topfeatures(dfm_pt1NP)
  mas ciego    si  dios   ser   tan   día   tal  vino  pues 
   40    33    33    30    28    27    26    25    24    23 
> topfeatures(dfm_pt2NP)
  amo  bien señor   mas  dios    si  casa  pues  ansí   día 
   57    55    55    48    46    41    37    31    27    26 
> topfeatures(dfm_pt1NP, 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:



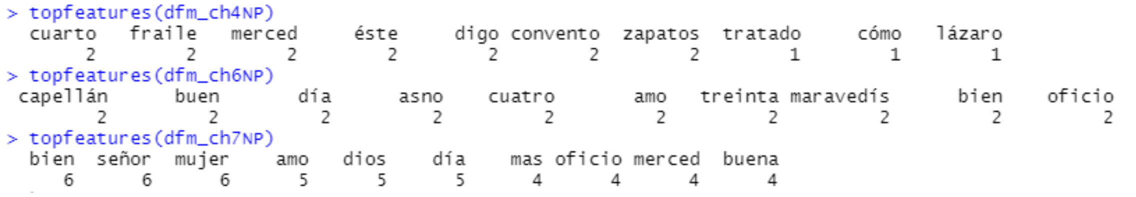
```
> topfeatures(dfm_ch1NP)
ciego   mas  vino   mal    si   tal  dios   ser   tan  pues 
   28    21    20    15    15    14    13    13    13    12 
> topfeatures(dfm_ch2NP)
  mas    si  dios   día  arca   ser   tan noche llave   amo 
   19    18    17    16    16    15    14    14    13    13 
> topfeatures(dfm_ch3NP)
  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?



```
> 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”.