Comparing texts

Rodrigo Esteves de Lima-Lopes

Purpose of this notebook

In this post I am going to discuss some strategies of comparison between texts. It was produced in order to assist colleagues who work in the area of Corpus Linguistics and Systemic Functional Linguistics, as a way to use R in their research. It is part of my CNPq-funded project and seeks to make corpus tools and network analysis accessible. If you have any doubts or wish to make any research contact, please send me an email.

This document is based mostly in the package Tidytext and in the following book:

Silge, Julia and David Robinson. 2017. Text Mining with R: A Tidy Approach. First edition. Beijing/Boston: O'Reilly.

Clarisse Lispector

In this notebook I am going to compare two novels by Clarisse Lispector: A Hora da Estrela and A Paixão segundo GH in their original Portuguese version. Some more information about Lispector might be found here.

Loading R packages

In this article we are going to use the following packages:

```
library(tidytext)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(readr)
library(tidyverse)

## ── Attaching packages ──────────────────────────────────────────────────────────────────────────────────────────────
tidyverse 1.2.1 ──
```

```
## ✓ ggplot2 3.1.1     ✓ purrr   0.3.2
## ✓ tibble  2.1.1     ✓ stringr 1.4.0
## ✓ tidyr   0.8.3     ✓ forcats 0.4.0

## ── Conflicts
─────────────────────────────────────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()

library(tm)

## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':
##
##     annotate

library(tidyr)
library(scales)

##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

Each of these packages has a specific function:

1. tidytext: Text manipulation
2. dplyr: Data manipulation, such as tables and data frames
3. readr: Reads documents and data into R
4. tidyverse: Loads some functions from other packages
5. tm: Text manipulation and corpus creation
6. tidyr: Data manipulation, such as tables and data frames
7. scales: Data representation

Data and stop-words

Since my focus here is on the semantically relevant Lexis, I will load a list of stopwords, in order to filter prepositions, articles and conjunctions. There are a number of stop-words list for Portuguese, I would advise you to prepare your own based on your research objectives. The code bellow also loads the two novels in R.

```
stopwords <- read_csv("stop_port2.csv")

## Parsed with column specification:
## cols(
```

```
##   word = col_character()
## )

estrela <- readLines("estrelab.txt")
paixao <- readLines("paixaob.txt")
```

The analysis

Create a wordlist

Our fist step will be processing the two novels in a set of words with their frequencies.

```
estrela <- data.frame(text = estrela, stringsAsFactors = F)
estrela.tidy <- estrela %>%
  unnest_tokens(word, text)%>%
  anti_join(stopwords)

## Joining, by = "word"

paixao <- data.frame(text = paixao, stringsAsFactors = F)
paixao.tidy <- paixao %>%
  unnest_tokens(word, text)%>%
  anti_join(stopwords)

## Joining, by = "word"
```

In the code above:

1. The first line creates a data frame from each novel. It is important to make the files processable by R.
2. The following lines process the data frame in order to:

- Make it in a word per line format
- Filter stop-words out.

Here is an example of how it reads:

```
head(paixao.tidy)

##         word
## 1 procurando
## 2 procurando
## 3   tentando
## 4   entender
## 5   tentando
## 6     alguem
```

As we can see, there is a column which displays a sequential number, representing the line of the list, and another which displays the words itself. However it is also possible to observe that procurando is present twice in the list. It happens because we are not counting the occurrences, but only displaying the novel's words.

```
paixao.l<- paixao.tidy%>%
  count(word, sort = TRUE)
esterla.l <- estrela.tidy%>%
  count(word, sort = TRUE)
```

The code above transforms these lists in something similar to a wordlist. The first line creates the file itself, which I chose to name with the suffix "*.l". The operator "%>%" is part of the tidyverse, its function is to help us to "pass" one line of code through to the other and perform more than a command at once. The following code counts the words and creates a new column with the results.

```
head(paixao.l)
```

```
## # A tibble: 6 x 2
##   word       n
##   <chr>  <int>
## 1 mim      227
## 2 vida     193
## 3 barata   180
## 4 amor     111
## 5 medo      87
## 6 mundo     87
```

```
head(esterla.l)
```

```
## # A tibble: 6 x 2
##   word         n
##   <chr>    <int>
## 1 macabea     91
## 2 vida        81
## 3 voce        79
## 4 moca        54
## 5 historia    52
## 6 disse       47
```

Our next step is to build a table comparing the frequency of such words. It is necessary in order to generate the visualisations and make the analysis possible. In the first line we create a new object and add two columns in it, one for each book, and make sure that the list of words identify to the book where it is from. I the following line we clear the text, keeping only alphabetical words and deleting numbers and special symbols. We group the words by book (livro in Portuguese) and calculate the relative proportion of the words within each novel. Since absolute values do not mean much, we delete them.

```
frequencia.clarisse <- bind_rows(mutate(paixao.tidy, livro = "P"),
                                 mutate(estrela.tidy, livro = "H")) %>%
  mutate(word = str_extract(word, "[a-z']+")) %>%
  count(livro, word) %>%
  group_by(livro) %>%
  mutate(proportion = n / sum(n)) %>%
  select(-n) %>%
  spread(livro, proportion)
```

As we can see (below) the columns P and H represent each book. The numbers are the relative frequency and the term NA means that some words do not occur in a given novel. The list goes on for over 7,000 words.

```
frequencia.clarisse
```

```
## # A tibble: 7,386 x 3
##    word              H           P
##    <chr>         <dbl>       <dbl>
##  1 aaaar      0.000117 NA
##  2 abafada    0.000117 NA
##  3 abafado    NA          0.0000616
##  4 abafar     NA          0.0000616
```

```
##  5 abafava    0.000117 NA
##  6 abaixando NA          0.000123
##  7 abaixara  NA          0.0000616
##  8 abaixava   0.000117   0.0000616
##  9 abaixei   NA          0.000123
## 10 abaixo    NA          0.000123
## # … with 7,376 more rows
```

Visualising the data

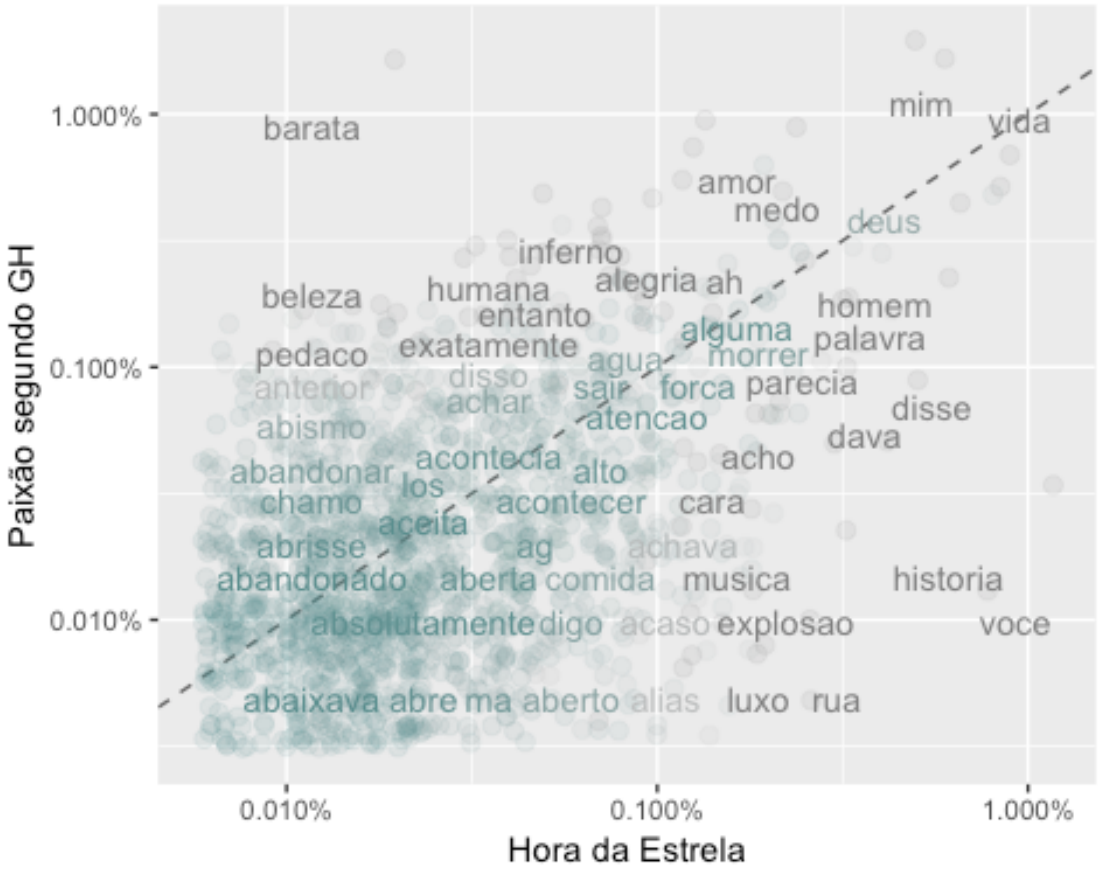
Now it is time to compare the Lexis in the two books. We will do so by using a ggplot command. I have to admit that visualisations is my poorest skill in R, so pardon me for my lack of ability in explaning the code bellow.

Mostly we are putting both novels side by side, making them overlap. The position is calculated my a log scale that helps to represent the overlapping. The more central a word is in the graphic, the less "specific" of a book it is.

```
ggplot(frequencia.clarisse, aes(x = H, y = P,
                      color = abs(H - P))) +
  geom_abline(color = "gray40", lty = 2) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.3, height = 0.3) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  scale_color_gradient(limits = c(0, 0.001),
                       low = "darkslategray4", high = "gray75") +
  theme(legend.position="none") +
  labs(y = "Paixão segundo GH", x = "Hora da Estrela")

## Warning: Removed 5919 rows containing missing values (geom_point).

## Warning: Removed 5920 rows containing missing values (geom_text).
```


1.000%
0.100%
0.010%
Paixão segundo GH
0.010%
0.100%
1.000%
Hora da Estrela
barata
mim
vida
amor
medo
deus
inferno
alegria
ah
beleza
humana
entanto
homem
alguma
palavra
pedaco
exatamente
agua
morrer
anterior
disso
sair
forca
parecia
achar
disse
atencao
abismo
dava
acho
abandonar
acontecia
alto
los
chamo
acontecer
cara
aceita
abrisse
ag
achava
abandonado
aberta
comida
musica
historia
absolutamente
digo
acaso
explosao
voce
abaixava
abre
ma
aberto
alias
luxo
rua