"The Sorrows of Young Werther" a Word Frequency Analysis

Juan José Alvarado

2025-04-22

Introduction

What benefit may bring analyzing how many times a word is repeated? What does it mean that some words are repeated most or less? This type of project, even though it seems introductory, lays the foundation of several crucial concepts when it comes to analyzing data. Have you wondered how a computer can understand, interpret and generate human language? Well let me tell your it starts right here!

Language is a massive source of data so being able to process and analyse it is fundamental. Comprehension of text structure, key words and topics are high regarded skills in most of industries: social media comments, classify documents with automate processes, optimize data searching engines precision are just a few.

Follow along this short project and discover a little bit more about word processing! I chose a light short novel called "The Sorrows of Young Werther" by J. W. Von Göthe. I downloaded the *html* version and converted to a *.txt file*, you can find it here: The Project Gutenberg

What should we expect?

When we think and express ideas, emotions, sensations through speaking or writing we would expect to use words related to the topic we have on our minds. If you have food on your mind, words like: kitchen, restaurant, vegetables or fruits most likely be the ones you use. But as we know, humans have a great talent in using context to give the words a completely different meaning so it is important to be aware of this when analyzing and making conclusions.

"The Sorrows of Young Werther" is a epistolary novel (a novel written as a collection of letters between fictional characters) where the main topic is one-sided love and how it could affect a young man´s life. We could expect to find in our analysis a repetition of words related to strong emotions of excessive love and also great sadness and pain.

Lets continue and find out!

Word Frequency Analysis

Text Loading and Manipulation

Firs of all we need the following packages to manipulate, filtrate and visualize character data:

```
# Loading Packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(stopwords)
```

```
## Warning: package 'stopwords' was built under R version 4.4.3
```

And then we need to read the text file in R. This will create a vector of text strings where each element represents a line of the file.

```r
# Loading Text
raw_text <- readLines("The Sorrows of Young Werther.txt", encoding = "UFT-8")
```

Specifying the type of encoding to "UFT-8" will make easier the special character management.

Now we have to process the text before making any word frequency calculations:

```r
# We need all the lines in one string to manipulate the text easier
text_unified <- paste(raw_text, collapse = " ")

# All Lowercase so the same word with upper case doesn't count as different
text_lower <- tolower(text_unified)

# Remove Punctuation as they don't count as words or part of them
text_no_punct <- str_replace_all(text_lower, "[[:punct:]]", "")

# Divide the text into individual words. "\\s+" is an expression that matches empty spaces
words <- unlist(str_split(text_no_punct, "\\s+"))
```

Frequency Calculation

Commonly used words in language like "a", "the,"is", "and", are often excluded from text processing simply because they don't carry much distinct information about the content of the text. So we have to get rid of them: the package *stopwords* includes a list of this words and in many languages.

```r
# Create a list of stopwords (in this case in English)
stopwords <- stopwords("en")

# Filtrate the processed text to not include the list of words you created
no_stopwords <- words[!words %in% stopwords]
```

Also is important to take care of possible empty lines

```r
# Possible empty lines
no_stopwords <- no_stopwords[no_stopwords != ""]
```

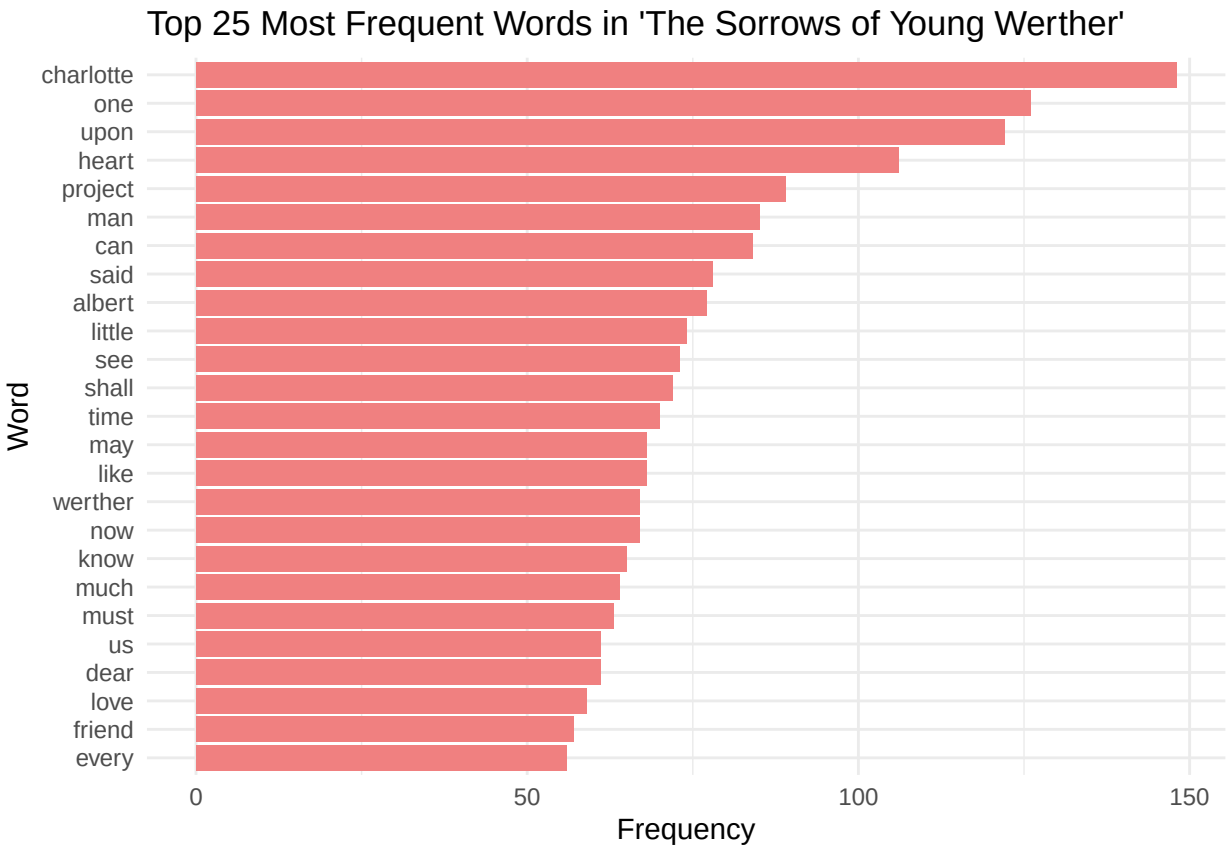
It is time to make the frequency calculations!

```r
# Word Frequency Calculation
word_freq <- table(no_stopwords)
df_freq <- as.data.frame(word_freq)
colnames(df_freq) <- c("Words", "Frequency")

# Order by Frequency
df_ordered <- df_freq[order(df_freq$Frequency, decreasing = TRUE), ]
```

The "table()" function calculates the frequency of each word of inside the vector. We converted the table into a data frame with "as.data.frame()"; gave names to the columns of the data frame. Lastly order the data frame by frequency in decreasing mode to get the most frequent at the top.

Now is just a matter of visualizing the data...

```r
top_words <- 25 #Could be any number you like

ggplot(df_ordered[1:top_words, ], aes(x = reorder(Words, Frequency), y = Frequency)) +
  geom_bar(stat = "identity", fill = "lightcoral") +
  coord_flip() +
  labs(title = paste("Top", top_words, "Most Frequent Words in 'The Sorrows of Young Werther'"),
       x = "Word",
       y = "Frequency") +
  theme_minimal()
```



There you go!

Conclusions

1. It comes as a surprise to many, including me, that in the *top 25 repeated words* used in this one-side love story, words like: friend, love, dear are at the bottom.
2. It is true that the word "charlotte" who is the character young Werther loves so deeply is the most repeated in the whole novel and just 3 positions below it, the word "heart" appears.
3. Most of the words (adjectives, adverbs, subjects) don´t seem to relate to a particular topic, meaning the context is more important to the author in this novel than specific words.

As you can see through this short project, extracting this kind of information from text files facilitates the interpretation of language data that can used in many industries and studies.

You can do more intricate exercises like comparing to different authors of the same genera and see how they can approach the same topic. Or compare two texts of the same author to study the evolution in his writing, etc. Just be curious and try it!