Data scraping from YouTube Channels

Rodrigo Esteves de Lima-Lopes
University of Campinas
rll307@unicamp.br

Contents

1 Introduction

This script was developed for the analysis of Portuguese. I hope it helps colleagues in the LC area and popularize the use of R. It is part of our research project developed with CNPQ. Please drop me a line if you have any doubts or need any help.

2 Purpose of this repository

This git brings the scripts for my article:

- Lima-Lopes R.E. (forthcoming). Beyond the Binary: Trans Women's Video Activism on YouTube. Accepted for publication at *Digital Scholarship in the Humanities.*

This script is specifically about **data scraping**.

3 Packages

For data scraping and manipulation we are going to need some packages, each has a different function

- abjutils: diacritic removal in Brazilian Portuguese
- tm, tidytext, tidyverse, magrittr: data manipulation and cleaning
- ggridges: graph plotting
- formattable: table formating
- reticulate: interface between R and Python

Please, note that <`your Python instalation`> refers to your Python executable path.

```
library(abjutils)
library(tidytext)
library(reticulate)
reticulate::use_python("<your Python instalation>", required = TRUE)
library(tidyverse)
library(magrittr)
library(stm)
library(tm)
library(ggridges)
library(formattable)
options(scipen = 999)
```

4 Creating and running the basic command

Those are the fields we are going to use for scraping the data from YouTube channels.

```
basic.fields <- c("id", "title", "alt_title", "creator", "release_date",
                "timestamp", "upload_date", "duration", "view_count",
                "like_count", "dislike_count", "comment_count")
```

Now we are going to format the fields, using &&& as separators

```
fields <- fields_raw %>%
  map_chr(~paste0("%(", ., ")s")) %>%
  # use &&& as fiels separator
  paste0(collapse = "&&&") %>%
  # add quotes in the beging and end of each stream
  paste0('"', ., '"')
```

The next variable is the link for the video or channel you intend to research. Please, note that due to ethical reasons I cannot provide the links I used for the article.

```
url <- "<your channel or video URL>"
```

Next, let us make the query command

```
cmd_raw <- str_glue("youtube-dl -o {fields} -i -v -w --skip-download --write-auto-sub --sub-lang pt
                    --sub-format vtt {url}")
```

Now we need to tell the system where the subtitles will be saved

```
Captions.Folder <- "<path to the folder>" # Informs the folder location
fs::dir_create(Captions.Folder) # Creates the folder
download.captions <- str_glue("cd {Captions.Folder} && {cmd_raw}") # Creates the actual command
```

Finally we run the dollowing command and wait.

```
system(download.captions)
```