Scraping YouTube captions

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

Contents

1 Introduction **1**
1.1 What we need 1

2 Creating the base command **1**

1 Introduction

This script brings a fast solution to download YouTube captions Here we are going to integrate R, Python and the base system in a very elegant script.

This script is largely inspired by the work of Silas Gonzaga, to whom all the credit should be given.

1.1 What we need

Some R packages are important for our exercise:

- `reticulate` - Python integration
- `tidyverse` - Data manipulation
- `formattable` - Table manipulation
- `purrr` - Function mapping
- `magrittr` - for the `%>%` operator
- `jsonlite` - for the manipulation of json files
- `stringr` - for string manipulation and command building

Outside R we also need:

- A working installation of `Python`
- A working installation of `youtube-dl`.
- A working installation of `Webvtt-py`

2 Creating the base command

Like in the last script, we will create a command to integrate with the system

1) Define the fields

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

2) Define the fields to be applied to the command

```
fields <- fields_raw %>%
  map_chr(~paste0("%(", ., ")s")) %>%
  # use &&& as field separator
  paste0(collapse = "&&&") %>%
  # add quotation marks at the end and begging of the string
  paste0('"', ., '"')
```

3) Define the channel URL. Please note it will work with both channels or individual videos

```
channel_url <- "https://www.youtube.com/channel/UCynXCso-wU6E4V-DnsHU7mA"
```

4) Create the query. Please note that the `str_glue`command makes the *{}* elements to change when the variable changes.

```
cmd_ytdl <- str_glue("youtube-dl -o {fields} -i -v -w --skip-download --write-auto-sub --sub-lang pt --
```

5) Now we create the final query and the folder where the captions will be downloaded

```
Sub.folder <- "subtitles"
fs::dir_create(Sub.folder)

cmd <- str_glue("cd {Sub.folder} && {cmd_ytdl}")
system(cmd)
```

- The `fs::dir` creates the directory I declared at `Sub.folder`
- The `system` executes the command on my system.

6) For cleaning the data we will need

- Map from the system the directory where my files R
- Source a Python script for data cleaning
- Source a R file with some useful functions

```
my.captions <- dir(Sub.folder, pattern = '*.vtt', full.names = TRUE)
source_python("caption_to_vector.py")
source("functions.R")
```

7) Finally we clean and save it all in a data frame

```
df <- my.captions %>%
  map_df(caption_to_df) %>%
  select(-comment_count) %>%
  mutate(upload_date = lubridate::ymd(upload_date)) %>%
  mutate_at(vars(duration:dislike_count), as.numeric)
```