Automatic Sampling and Analysis of YouTube Data

Excursus: Retrieving Video Subtitles

Julian Kohne
Johannes Breuer
M. Rohangis Mohseni

2022-02-22

Retrieving *YouTube* Video Subtitles

- Instead of transcribing a video, you can retrieve its subtitles via the *YouTube* API
- What research would you conduct with video subtitles?

Types of *YouTube* Subtitles

- Videos with automatically created subtitles (*ASR*)
  - Always in English, even if video language is not English
  - Can be downloaded, but text quality can be bad (especially if translated)
- Videos without any subtitles
  - Not sure if even possible because there always seems to be an *ASR*
- Videos with more than one set of subtitles
  - Examples: *ASR* and regular subtitle, more than one language, more than one subtitle for the same language
  - Can be downloaded, but subtitle for analysis must be selected

Disclaimer

Due to a change to the *YouTube* API, the `tuber` function for retrieving video subtitles only work for videos that were created with the same account as the app used for the API access (see this closed `tuber` issue on GitHub). We will still discuss this function, but recommend that you use the `youtubecaption` package for collecting subtitles for videos that you have not created yourself.

Retrieving Video Subtitles with `tuber`

First, we need to get the list of subtitles for a video.

```
library(tuber)

caption_list <- list_caption_tracks(video_id = "nI_0fkQOG6Q")
```

Note: The `tuber` function `list_caption_tracks()` has an API quota cost ~ 50.

Retrieving Video Subtitles with `tuber`

Next, we need to get the ID of the subtitles we want to collect.

```
ID <- caption_list[1,"id"]
```

Note: You can adapt the number to select the subtitle that you want. The 1 in this example corresponds to the *ASR* captions (ASR = automatic sub).

Retrieving Video Subtitles with `tuber`

After that, we need to retrieve the subtitles and convert them from raw to char.

```
text <- rawToChar(get_captions(id = ID, format = "sbv"))
```

Now we can save the subtitles to a subtitle file.

```
write(text, file = "Captions.sbv", sep="\n")
```

Converting Subtitles

- Subtitles come in a special format called SBV
- The format contains time stamps etc. that we do not need for text analysis
- We can read the format with the package `subtools`

Converting Subtitles

```r
remotes::install_github("fkeck/subtools")
library(subtools)

subs <- read_subtitles("Captions.sbv", format = "subviewer")
```

With `subtools`, we can also retrieve the text from the subtitles.

```r
subtext <- get_raw_text(subs)
```

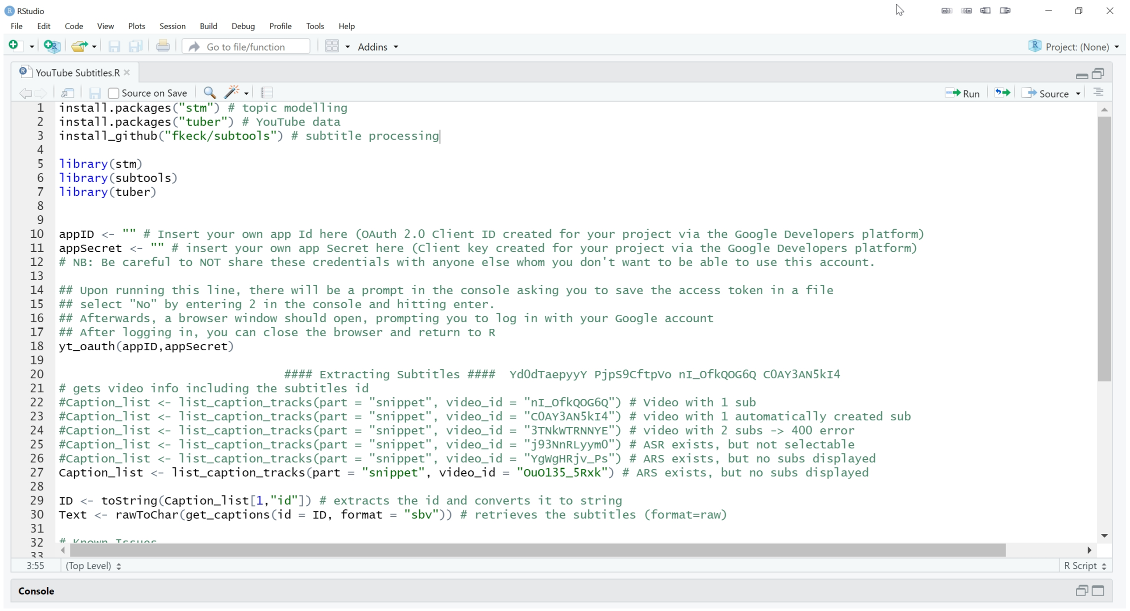
Now the text is ready for further analysis (see the previous sessions for examples).

Retrieving Video Subtitles with `youtubecaption`

- Alternatively, you can retrieve captions with the package `youtubecaption`
- **Pros**:
  - No credentials necessary, therefore no quota reduction (package uses web scraping)
  - Subtitles are automatically converted into a dataframe, including texts and timestamps, so no manual conversion is needed
- **Cons**:
  - If there is more than one subtitle version per language, there is no way to select a specific one in `R` (by default, the manually generated ones are selected; if you want the ASR subtitles instead, you would need to adapt the underlying `Python` script)
  - You need to install *Anaconda*

Demo: Collecting & processing subtitles with `tuber` & `subtools`

You can find the code for collecting subtitles for *YouTube* videos in the `YouTubeSubtitles.R` file in the `scripts` folder.



```r
install.packages("stm") # topic modelling
install.packages("tuber") # YouTube data
install_github("fkeck/subtools") # subtitle processing

library(stm)
library(subtools)
library(tuber)


appID <- "" # Insert your own app Id here (OAuth 2.0 Client ID created for your project via the Google Developers platform)
appSecret <- "" # insert your own app Secret here (Client key created for your project via the Google Developers platform)
# NB: Be careful to NOT share these credentials with anyone else whom you don't want to be able to use this account.

## Upon running this line, there will be a prompt in the console asking you to save the access token in a file
## select "No" by entering 2 in the console and hitting enter.
## Afterwards, a browser window should open, prompting you to log in with your Google account
## After logging in, you can close the browser and return to R
yt_oauth(appID,appSecret)

                                #### Extracting Subtitles ####  YdOdTaepyyY PjpS9CftpVo nI_OfkQOG6Q COAY3AN5kI4
# gets video info including the subtitles id
#Caption_list <- list_caption_tracks(part = "snippet", video_id = "nI_OfkQOG6Q") # Video with 1 sub
#Caption_list <- list_caption_tracks(part = "snippet", video_id = "COAY3AN5kI4") # video with 1 automatically created sub
#Caption_list <- list_caption_tracks(part = "snippet", video_id = "3TNkWTRNNYE") # video with 2 subs -> 400 error
#Caption_list <- list_caption_tracks(part = "snippet", video_id = "j93NnRLyym0") # ASR exists, but not selectable
#Caption_list <- list_caption_tracks(part = "snippet", video_id = "YgWgHRjv_Ps") # ARS exists, but no subs displayed
Caption_list <- list_caption_tracks(part = "snippet", video_id = "OuO135_5Rxk") # ARS exists, but no subs displayed

ID <- toString(Caption_list[1,"id"]) # extracts the id and converts it to string
Text <- rawToChar(get_captions(id = ID, format = "sbv")) # retrieves the subtitles (format=raw)

```

Our Recommendation for Collecting Subtitels

If you want to collect subtitles for videos that have been created/uploaded with an account that you have or can easily get access to (e.g., from your institution or a project partner), you can use the functions from the `tuber` package as shown in the demo script.

If you want to collect subtitles from other videos, we suggest that you use the `list_caption_tracks()` function from the `tuber` package to check which subtitles are available and the `get_caption` function from the `youtubecaption` package for collecting them (given that you can properly set up *Anaconda* for using this package).

Any (further) questions?