

Automatic Sampling and Analysis of YouTube Data

Excursus: Retrieving Video Subtitles

Julian Kohne
Johannes Breuer
M. Rohangis Mohseni

2021-02-25

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 recent change to the *YouTube* API, the `tuber` function for retrieving video subtitles seems to 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`

- Retrieve a list of subtitles with
 - `tuber::list_caption_tracks()`
- Quota costs ~ 50

Retrieving Video Subtitles with `tuber`

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

```
caption_list <- list_caption_tracks(video_id =  
"nI_OfkQOG6Q")
```

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

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

- Adapt the number to select the subtitle that you want (ASR = automatic sub)
- 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`

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

- With subtools, we can also retrieve the text from the subtitles

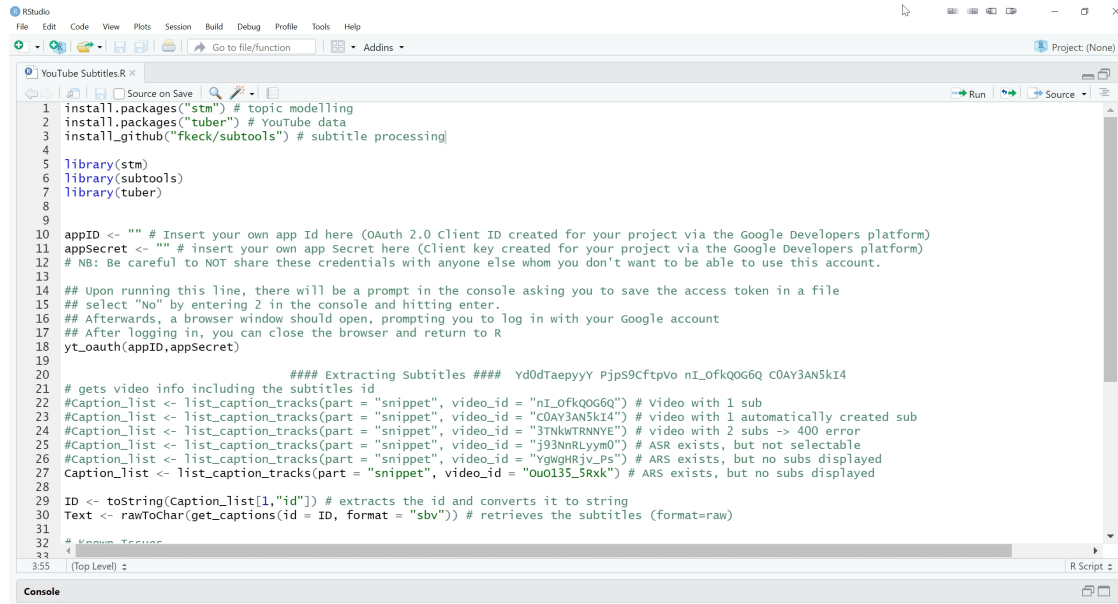
```
subtext <- get_raw_text(Subs)
```

- Now the text is ready for text analysis

Retrieving Video Subtitles with youtubecaption

- Alternatively, you can retrieve captions with the package `youtubecaption`
- **Pros:**
 - No credentials necessary, therefore no quota reduction
 - 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
 - You need to install *Anaconda*

Time for a Short Live Demo



```

1 install.packages("stm") # topic modelling
2 install.packages("tuber") # YouTube data
3 install_github("fkeck/subtools") # subtitle processing
4
5 library(stm)
6 library(subtools)
7 library(tuber)
8
9
10 appID <- "" # Insert your own app ID here (OAuth 2.0 client ID created for your project via the Google Developers platform)
11 appSecret <- "" # insert your own app secret here (Client key created for your project via the Google Developers platform)
12 # NB: Be careful to NOT share these credentials with anyone else whom you don't want to be able to use this account.
13
14 ## Upon running this line, there will be a prompt in the console asking you to save the access token in a file
15 ## select "No" by entering 2 in the console and hitting enter.
16 ## Afterwards, a browser window should open, prompting you to log in with your Google account
17 ## After logging in, you can close the browser and return to R
18 yt_oauth(appID,appSecret)
19
20 ##### Extracting Subtitles ##### Yd0dTaepyy PjpS9CftpVo nI_ofkQOG6Q C0AY3AN5kI4
21 # gets video info including the subtitles id
22 #Caption_list <- list_caption_tracks(part = "snippet", video_id = "nI_ofkQOG6Q") # Video with 1 sub
23 #Caption_list <- list_caption_tracks(part = "snippet", video_id = "C0AY3AN5kI4") # video with 1 automatically created sub
24 #Caption_list <- list_caption_tracks(part = "snippet", video_id = "3TNkWTRNNYE") # video with 2 subs -> 400 error
25 #Caption_list <- list_caption_tracks(part = "snippet", video_id = "j93NnRLyym0") # ASR exists, but not selectable
26 #Caption_list <- list_caption_tracks(part = "snippet", video_id = "YgwgHRjv_Ps") # ASR exists, but no subs displayed
27 Caption_list <- list_caption_tracks(part = "snippet", video_id = "Ou0l35_5Rxk") # ASR exists, but no subs displayed
28
29 ID <- toString(Caption_list[1,"id"]) # extracts the id and converts it to string
30 Text <- rawToChar(get_captions(id = ID, format = "sbv")) # retrieves the subtitles (format=raw)
31
32 # Known Issues
33
34
35

```

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

Any (further) questions?