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

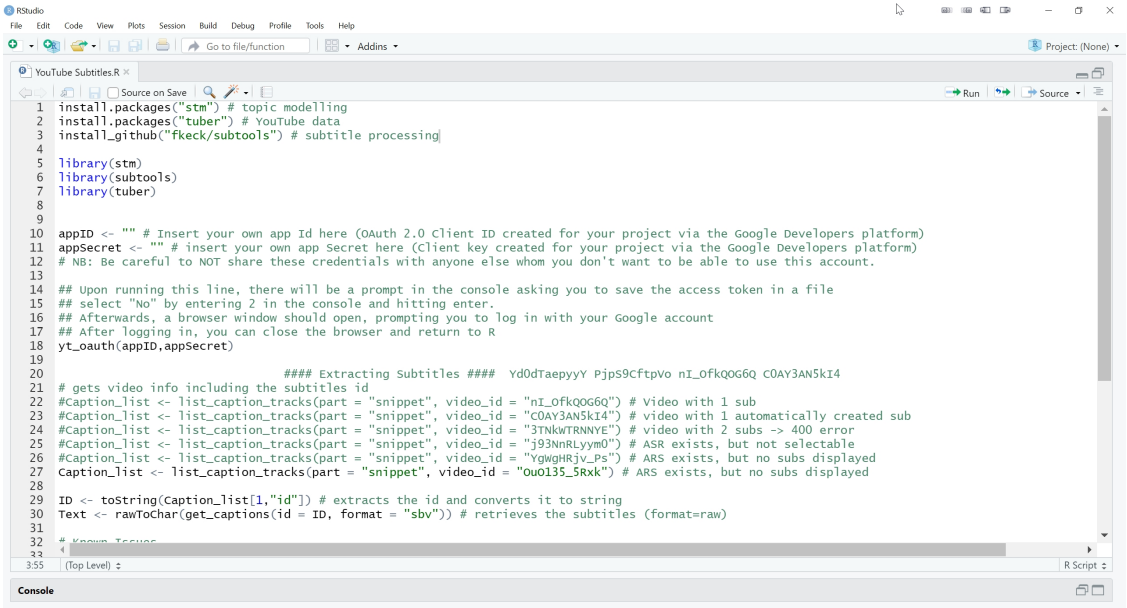
```
subtext <- get_raw_text(Subs)
```

- 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



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

Any (further) questions?