Collecting and Analyzing Social Media Data

Tiago Ventura | Center for Social Media and Politics | NYU

Big Data for Development and Governance

10/21/2022

Collecting and Analyzing Youtube Data

What I am presenting here follows very closely a notebook using the Python Library youtube-data-api developed by Megan Brown, Senior Engineer at the Center for Social Media and Politics at NYU, and some other colleagues.

Thanks Megan!

What kind of data can you get from the Youtube API?

Youtube has a very extensive api. There are a lot of data you can get access to.

See a comprehensive list here

What is included in the package:

- video metadata
- channel metadata
- playlist metadata
- subscription metadata
- featured channel metadata
- comment metadata
- search results

Installing

```
# run in the command line
pip install youtube-data-api
```

How to get an API key

A quick guide: https://developers.google.com/youtube/v3/getting-started

- You need a Google Account to access the Google API Console, request an API key, and register your application. You can use your GMail account for this if you have one.
- Create a project in the Google Developers Console and obtain authorization credentials so your application can submit API requests.
- After creating your project, make sure the YouTube Data API is one of the services that your application is registered to use.

Calling packages

```
# call some libraries
import os
import datetime
import pandas as pd
pd.set_option('display.max_columns', None)
```

```
# pass your keys
from youtube_api import YouTubeDataAPI
from youtube_api.youtube_api_utils import *
from dotenv import load_dotenv

# load keys from  environmental var
load_dotenv() # .env file in cwd
```

```
## True
```

```
api_key = os.environ.get("YT_KEY")
```

Create a Python Client to interact with the API

```
# create a client
yt = YouTubeDataAPI(api_key)
```

Starting with a Channel

Let's start with the `LastWeekTonight` channel

https://www.youtube.com/user/LastWeekTonight

First we need to get the channel id

```
channel_id = yt.get_channel_id_from_user('LastWeekTonight')
print(channel_id)
```

```
## UC3XTzVzaHQEd30rQbuvCtTQ
```

Channel metadata

```
# collect metadata
channel_metadata = yt.get_channel_metadata(channel_id)
pd.DataFrame([channel_metadata]).head()
```

```
##                  channel_id            title  account_creation_date keywords  \
## 0  UC3XTzVzaHQEd30rQbuvCtTQ  LastWeekTonight           1.395179e+09     None
##
##                                          description  view_count video_count  \
## 0  Breaking news on a weekly basis. Sundays at 11...  3474260272         400
##
##   subscription_count playlist_id_likes       playlist_id_uploads  \
## 0            9070000                    UU3XTzVzaHQEd30rQbuvCtTQ
##
##                                          topic_ids country  \
## 0  https://en.wikipedia.org/wiki/Society|https://...    None
##
##              collection_date
## 0 2022-10-19 11:29:03.272270
```

Subscriptions of the channel.

```
subs = yt.get_subscriptions(channel_id)
pd.DataFrame(subs).head()
```

```
##   subscription_title   subscription_channel_id subscription_kind  \
## 0          trueblood  UCPnlBOg4_NU9wdhRN-vzECQ   youtube#channel
## 1      GameofThrones  UCQzdMyuz0Lf4zo4uGcEujFw   youtube#channel
## 2                HBO  UCVTQuK2CaWaTgSsoNkn5AiQ   youtube#channel
## 3          HBOBoxing  UCWPQB43yGKEum3eW0P9N_nQ   youtube#channel
## 4            Cinemax  UCYbinjMxWwjRpp4WqgDqEDA   youtube#channel
##
##    subscription_publish_date            collection_date
## 0               1.395357e+09 2022-10-19 11:29:04.142185
## 1               1.395357e+09 2022-10-19 11:29:04.142232
## 2               1.395357e+09 2022-10-19 11:29:04.142269
## 3               1.395357e+09 2022-10-19 11:29:04.142304
## 4               1.424812e+09 2022-10-19 11:29:04.142339
```

List of videos of the channel

You first need to convert the `channel_id` into a playlist id to get all the videos ever posted by a channel using a function from the `youtube_api_utils` in the package.

```
from youtube_api.youtube_api_utils import *
playlist_id = get_upload_playlist_id(channel_id)
print(playlist_id)
```

```
## UU3XTzVzaHQEd30rQbuvCtTQ
```

```
## Get video ids
videos = yt.get_videos_from_playlist_id(playlist_id)
df = pd.DataFrame(videos)
df.head()
```

```
##       video_id                channel_id  publish_date  \
## 0  Ns8NvPPHX5Y  UC3XTzVzaHQEd30rQbuvCtTQ  1.666003e+09
## 1  kCOnGjvYKI0  UC3XTzVzaHQEd30rQbuvCtTQ  1.665398e+09
## 2  eJPLiT1kCSM  UC3XTzVzaHQEd30rQbuvCtTQ  1.664793e+09
## 3  uySgklnlX3Y  UC3XTzVzaHQEd30rQbuvCtTQ  1.664188e+09
## 4  DNy6F7ZwX8I  UC3XTzVzaHQEd30rQbuvCtTQ  1.662979e+09
##
##              collection_date
## 0 2022-10-19 11:29:04.805890
## 1 2022-10-19 11:29:04.805924
## 2 2022-10-19 11:29:04.805952
## 3 2022-10-19 11:29:04.805978
## 4 2022-10-19 11:29:04.806003
```

Collect video metadata

Then you can get the video ids, and collect metadata, comments, among many others.

```
# id for videos as a list
df.video_id.tolist()[:5]
```

```
## ['Ns8NvPPHX5Y', 'kCOnGjvYKI0', 'eJPLiT1kCSM', 'uySgklnlX3Y', 'DNy6F7ZwX8I']
```

```
#grab metadata
video_meta = yt.get_video_metadata(df.video_id.tolist()[:5])

#visualize
pd.DataFrame(video_meta[:5])
```

```
##       video_id    channel_title                channel_id  video_publish_date  \
## 0  Ns8NvPPHX5Y  LastWeekTonight  UC3XTzVzaHQEd30rQbuvCtTQ        1.666003e+09
## 1  kCOnGjvYKI0  LastWeekTonight  UC3XTzVzaHQEd30rQbuvCtTQ        1.665398e+09
## 2  eJPLiT1kCSM  LastWeekTonight  UC3XTzVzaHQEd30rQbuvCtTQ        1.664793e+09
## 3  uySgklnlX3Y  LastWeekTonight  UC3XTzVzaHQEd30rQbuvCtTQ        1.664188e+09
## 4  DNy6F7ZwX8I  LastWeekTonight  UC3XTzVzaHQEd30rQbuvCtTQ        1.662979e+09
##
##                                          video_title  \
## 0  Transgender Rights II: Last Week Tonight with ...
## 1  Crime Reporting: Last Week Tonight with John O...
## 2  Museums: Last Week Tonight with John Oliver (HBO)
## 3  Bolsonaro: Last Week Tonight with John Oliver ...
## 4  Law & Order: Last Week Tonight with John Olive...
##
##                                   video_description video_category  \
## 0  John Oliver discusses the latest round of atta...             24
## 1  John Oliver discusses the outlets that cover c...             24
## 2  John Oliver discusses some of the world's most...             24
## 3  John Oliver discusses Jair Bolsonaro, how he's...             24
## 4  John Oliver discusses the wildly popular telev...             24
##
##   video_view_count video_comment_count video_like_count video_dislike_count  \
## 0          2344480                24895            98593                None
## 1          3251850                 5956            90164                None
## 2          4774576                12110           135467                None
## 3          4646414                14835           171157                None
## 4          5053055                 9073           119449                None
##
##                                    video_thumbnail video_tags  \
```

Collect Comments

```
ids = df.video_id.tolist()[:5]

# loop
list_comments = []
for video_id in ids:
  comments = yt.get_video_comments(video_id, max_results=10)
  list_comments.append(pd.DataFrame(comments))

# concat
df = pd.concat(list_comments)
```

```
df.keys()
```

```
## Index(['video_id', 'commenter_channel_url', 'commenter_channel_id',
##        'commenter_channel_display_name', 'comment_id', 'comment_like_count',
##        'comment_publish_date', 'text', 'commenter_rating', 'comment_parent_id',
##        'collection_date', 'reply_count'],
##       dtype='object')
```

```
df.head()
```

```
##       video_id                                commenter_channel_url  \
## 0  Ns8NvPPHX5Y  http://www.youtube.com/channel/UC33T1N9RE3fcVO...
## 1  Ns8NvPPHX5Y  http://www.youtube.com/channel/UCs0wh1A5wdsme-...
## 2  Ns8NvPPHX5Y  http://www.youtube.com/channel/UCrYqZMxVhwZm88...
## 3  Ns8NvPPHX5Y  http://www.youtube.com/channel/UCZsmzndxgRH_G-...
## 4  Ns8NvPPHX5Y  http://www.youtube.com/channel/UCrYqZMxVhwZm88...
##
##        commenter_channel_id commenter_channel_display_name  \
## 0  UC33T1N9RE3fcVOVengApc1A                          真OPJM
## 1  UCs0wh1A5wdsme-ac7A6DbGw                          Kesya
## 2  UCrYqZMxVhwZm88rrpFSGfCQ                      usakiwi1986
## 3  UCZsmzndxgRH_G-PSqH4ngGg                  Skyler Campbell
## 4  UCrYqZMxVhwZm88rrpFSGfCQ                      usakiwi1986
##
##                    comment_id  comment_like_count  comment_publish_date  \
## 0  UgxKgKUpJXFFlq4_Arp4AaABAg                   0          1.666208e+09
## 1  Ugzo2Ae21jNtIP6Yqet4AaABAg                   1          1.666207e+09
## 2  UgyqYBiXrGCfy0OLCNN4AaABAg                   0          1.666207e+09
## 3  Ugyr9IyRiZ0NVrYKz494AaABAg                   0          1.666207e+09
## 4  UgyaT4vkQXpDKKLjPxZ4AaABAg                   0          1.666207e+09
##
##                                                 text commenter_rating  \
## 0  Having done as I said, rolling around in the e...              none
## 1  We dont have gender pronounced here,\nNo past ...              none
## 2  I didn't care for the part saying God made me ...              none
## 3   Helping people play protect did not help anybody              none
## 4  Compare this med to opiods that have gotten so...              none
##
##   comment_parent_id            collection_date  reply_count
## 0              None 2022-10-19 11:29:08.735052            1
## 1              None 2022-10-19 11:29:08.735089            1
## 2              None 2022-10-19 11:29:08.735113            1
```

Related videos

Cool enough, the API allows you to get a sense (not perfect) of what YT recommend to users.

```
df = pd.DataFrame(yt.get_recommended_videos(ids[0]))
df.channel_title
```

```
## 0                 Saturday Night Live
## 1                         Kendall Rae
## 2                      LastWeekTonight
## 3    The Daily Show with Trevor Noah
## 4                      LastWeekTonight
## Name: channel_title, dtype: object
```

Search

The youtube API also allows you to search for most popular videos using queries.

```
df = pd.DataFrame(yt.search(q='urnas, fraude', max_results=10))
df.keys()
```

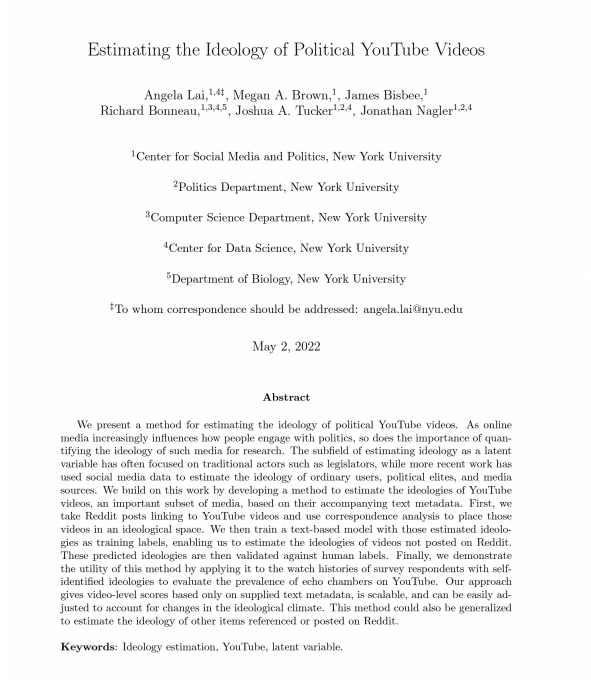
```
## Index(['video_id', 'channel_title', 'channel_id', 'video_publish_date',
##        'video_title', 'video_description', 'video_category', 'video_thumbnail',
##        'collection_date'],
##       dtype='object')
```

```
df[["channel_title", "video_title"]]
```

```
##            channel_title                                        video_title
## 0           vejapontocom  Giro VEJA | Alexandre de Moraes põe pressão so...
## 1                    UOL  Flávio Bolsonaro diz que não teve fraude nas u...
## 2         Jovem Pan News  Urna eletrônica: houve fraude no 1º turno? - B...
## 3      Rádio BandNews FM  Eleições: Resultados falsos da votação no exte...
## 4              CNN Brasil  Análise: Documento do PL de Bolsonaro aponta f...
## 5        Canal Nostalgia                 URNA ELETRÔNICA / Dá pra Hackear?
## 6       justicaeleitoral   FATO: Ter cópia do boletim de urna não é fraude
## 7                 RedeTV  Eleitores reclamam de fraude nas urnas durante...
## 8  Jornalismo TV Cultura  Eleições 2022: Confira momento em que Lula ult...
## 9            Record News  Bolsonaro diz que provará fraude nas urnas ele...
```

Want more about Youtube Data? Read these papers!

Lei et al, Estimating the Ideology of Political YouTube Videos



Estimating the Ideology of Political YouTube Videos

Angela Lai,[1,4‡], Megan A. Brown,[1], James Bisbee,[1]
Richard Bonneau,[1,3,4,5], Joshua A. Tucker[1,2,4], Jonathan Nagler[1,2,4]

[1]Center for Social Media and Politics, New York University

[2]Politics Department, New York University

[3]Computer Science Department, New York University

[4]Center for Data Science, New York University

[5]Department of Biology, New York University

[‡]To whom correspondence should be addressed: angela.lai@nyu.edu

May 2, 2022

Abstract

We present a method for estimating the ideology of political YouTube videos. As online media increasingly influences how people engage with politics, so does the importance of quantifying the ideology of such media for research. The subfield of estimating ideology as a latent variable has often focused on traditional actors such as legislators, while more recent work has used social media data to estimate the ideology of ordinary users, political elites, and media sources. We build on this work by developing a method to estimate the ideologies of YouTube videos, an important subset of media, based on their accompanying text metadata. First, we take Reddit posts linking to YouTube videos and use correspondence analysis to place those videos in an ideological space. We then train a text-based model with those estimated ideologies as training labels, enabling us to estimate the ideologies of videos not posted on Reddit. These predicted ideologies are then validated against human labels. Finally, we demonstrate the utility of this method by applying it to the watch histories of survey respondents with self-identified ideologies to evaluate the prevalence of echo chambers on YouTube. Our approach gives video-level scores based only on supplied text metadata, is scalable, and can be easily adjusted to account for changes in the ideological climate. This method could also be generalized to estimate the ideology of other items referenced or posted on Reddit.

Keywords: Ideology estimation, YouTube, latent variable.

Brown et al, Echo Chambers, Rabbit Holes, and Algorithmic Bias: How YouTube Recommends Content to Real Users