

# Media Insights

Darren Colby  
Jessup Jong

# Entry Point

Bash File

`./automate.sh`

`./slow_automate.sh`

Makefile

“make” or “make everything”

“make data/comment\_data.json”

“make data/clean\_comment\_data.json”

“make data/transcript\_data.json”

“make data/preprocessed\_comments.json”

# Video Upload Automation

```
python upload_video.py --file="videos/test1.mov"
```

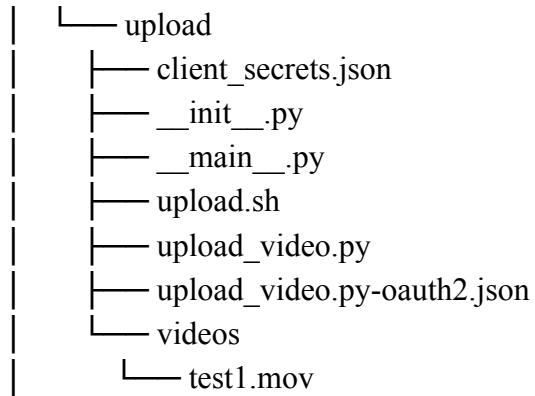
```
--title="Summer vacation in California"
```


```
--description="Had fun surfing in Santa Cruz"
```

```
--keywords="surfing,Santa Cruz"
```

```
--category="22"
```

```
--privacyStatus="public"
```








View channel on YouTube


Your channel


Media Insights Upload


 Dashboard


 **Content**

 Analytics

 Comments

 Subtitles

 Copyright

 Earn

Channel content









Videos

Live

Posts

Playlists

Filter

<input type="checkbox"/>	Video	Visibility	Restrictions	Date ↓	Views	Comments	L
<input type="checkbox"/>	<div><div>0:03</div></div> <div>Test Upload #1 Surfing in Santa Cruz</div>	 Public	None	Mar 6, 2023 Published	0	0	
<input type="checkbox"/>	<div><div>0:03</div></div> <div>Test Upload #0 Surfing in Santa Cruz</div>	 Public	None	Mar 6, 2023 Published	0	0	
<input type="checkbox"/>	<div><div>0:03</div></div> <div>Summer vacation in California Had fun surfing in Santa Cruz</div>	 Public	None	Mar 6, 2023 Published	0	0	
<input type="checkbox"/>	<div><div>0:03</div></div> <div>Summer vacation in California Had fun surfing in Santa Cruz</div>	 Unlisted	None	Mar 6, 2023 Uploaded	0	0	

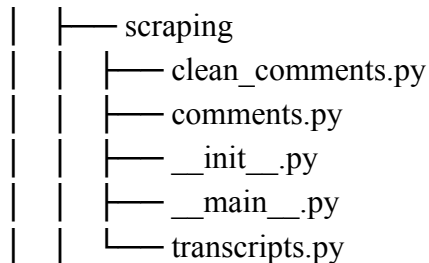
Authentication successful.  
Uploading file...  
Video id 'kLLB9trfxic' was successfully uploaded.

# Web Scraping: Transcripts

YouTube Transcript/Subtitle API (including automatically generated subtitles and subtitle translations)

Donate PayPal build passing coverage 100% license MIT pypi v0.5.0 python 3.5 | 3.6 | 3.7 | 3.8

transcript\_data.json



```
[["z-d0foxSkFU", "2mdAgnF0z0U", "Xo4txCMetIs", "eN0sAYQ3_NM", "crH_fqwKJuk"], [{"text": "buddy and some green beans", "start": 1.04, "duration": 3.9}, {"text": "[Music]", "start": 5.66, "duration": 7.48}, {"text": "that's gotta be 50 plus pounds of fat or", "start": 15.8, "duration": 6.72}, {"text": "no I did not just", "start": 23.64, "duration": 7.5}, {"text": "[Music]", "start": 25.61, "duration": 7.589}, {"text": "good", "start": 31.14, "duration": 0.59}, {"text": "[Music]", "start": 40.23, "duration": 3.07}, {"text": "did you get it", "start": 44.879, "duration": 2.601}, {"text": "I'm glad you go for a ride", "start": 49.86, "duration": 3.98}, {"text": "[Music]", "start": 64.379, "duration": 5.721}, {"text": "because there's no room", "start": 66.26, "duration": 3.84}, {"text": "oh", "start": 70.86, "duration": 2.6}, {"text": "I got the little girl", "start": 74.939, "duration": 3.261}, {"text": "this way", "start": 80.7, "duration": 3.14}, {"text": "that's my damn dog that's"}]
```

# Web Scraping: Comments

```
api_key = os.environ['API_KEY']
```

```
url = f"https://www.googleapis.com/youtube/v3/commentThrea
response = requests.get(url).json()
return response
```

```

"kind": "youtube#commentThreadListResponse",
"etag": "wJNSgrBIXj0nHeyCeFqHs1NhKx0",
"nextPageToken":
"QURTS19pMHV4U3lFWZw9fN01XdHrfa3A0dk5TM903THz0tW5DVFVUCy12cdhR2VsDUHx.
TRNTzQ0aEtaC6JJZ1g0UdhPVGZ6Xy1JtL2Z==",
"pageInfo": { "totalResults": 20, "resultsPerPage": 20 },
"items": [
{
"kind": "youtube#commentThread",
"etag": "7rKQmJRl7Xcnb4C9jKGw06DUUq0",
"id": "UgycG3LmbRzRx6FbQvX4AaABAg",
"snippet": {
"videoId": "w55xmZLwFbg",
"topLevelComment": {
"kind": "youtube#comment",
"etag": "B50JB8LMSKA4FNgjy1_0Xu8wAZoK",
"id": "UgycG3LmbRzRx6FbQvX4AaABAg",
"snippet": {
"videoId": "w55xmZLwFbg",
"textDisplay": "Why would you call them litter
mates? It's#39;s their siblings, all of them,
whether they's#39;re close or not, they's#39;re all
blood related. Therefore, they's#39;re siblings.",
"textOriginal": "Why would you call them litter
mates? It's their siblings, all of them, whether
they're close or not, they're all blood related.
Therefore, they're siblings.",

```

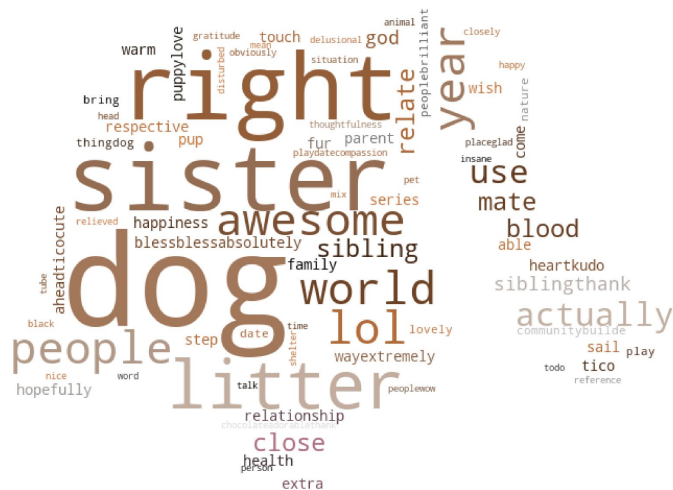
```
cleaned_comments = {}
for video_response in video_responses:
    cleaned_video = {}
    for i, comment in enumerate(video_response["items"]):
        try:
            text = comment["snippet"]["topLevelComment"]["snippet"]["textOriginal"]
            date = comment["snippet"]["topLevelComment"]["snippet"]["publishedAt"]
            cleaned_video += [(text, date)]
        except:
            print(f"Structure Different for Comment {i}")
    cleaned_comments[video_response["items"][0]["snippet"]["videoId"]] = cleaned_video
```

```
{
  "w55xmZLwFbg": [
    [
      "Why would you call them litter mates? It's their siblings, all of them",
      "2023-03-04T05:57:54Z",
    ],
    [
      "Thank you so much really god bless you \u2764\u2764\u2764",
      "2023-03-02T11:33:47Z",
    ],
    [
      "Bless \u2764\u2764\u2764",
      "2023-03-01T20:35:35Z",
    ],
    [
      "That\u2019s absolutely awesome. I smiled all of the way through.",
      "2023-03-01T14:33:12Z",
    ],
    [
      "ALL extremely AWESOME people, keeping a fur family in touch with the world",
      "2023-03-01T13:30:20Z",
    ],
    [
      "Kudos to Ernie's & Taco's pawrents for going that extra step for their babies",
      "2023-03-01T01:04:53Z",
    ],
  ],
}
```

# Word Cloud



# Frequency, shape and color



# Process Transcripts and Comments

## Pre-process transcript and comments

```
no_emojis = remove_emojis(text)
no_newline = re.sub(r'[\r\n\t]', '', no_emojis)
no_bracket = re.sub(r" ?\[^\)]+", "", no_newline)
no_punct = re.sub(r'^\w\s', '', no_bracket)
no_extra_space = re.sub(r'\s+', ' ', no_punct)
lowercase = no_extra_space.strip().lower()
no_digits = re.sub(r'[0-9]+', '', lowercase)

# Remove hyperlinks
no_links = re.sub(r'http\S+', '', no_digits)
no_links = re.sub(r"[!@#$]", '', no_links)
```

## Spell checking, translation, lemmatization

```
if fast:
    # Spell checking
    better_spelling = corrector(text)
else:
    # Translate to english
    sleep(1)
    translation = translators.translate_text(text)

    better_spelling = corrector(translation)

# Lemmatize and remove stopwords
doc = en_model(better_spelling)
clean_doc = " ".join([tok.lemma_ for tok in doc
                        if tok.lemma_ not in STOP_WORDS and len(tok.lemma_) > 1])

clean_comments.append(clean_doc); clean_dates.append(date)
```