Cross Culture Analysis

Vikki Sui (ks3747)

Introduction

The main goal of the research is to distinguish the cultural difference between China and United States through video. By watching videos, especially for people who can understand both languages, the difference of the two cultures might be obvious. However, we want to use the quantitative and analytic way to figure out if there is any difference specially on the attitude toward an event or the way of describing the event for the two countries.

Previous Work and Current Concentration

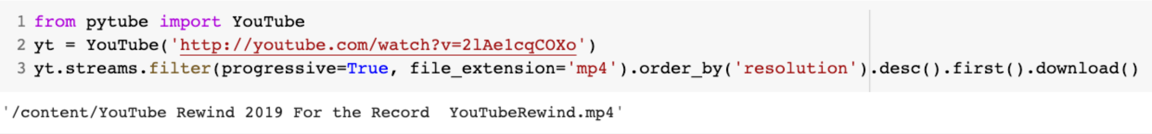
For previous semesters, the analysis was mainly on the topic of AlphaGo, the chess game event played by the world-best chess player and Google-developed AI. There has been existing video dataset and some analysis on this topic. I have read the report for previous semester. Since I only have the access to the videos and most of the old code for analysis does not work out well, I want to work on a new topic and collect a new dataset which is clearer than the existing dataset and the new dataset should contain more information that future analysis may need. Based on the special condition of 2020 and the huge amount of news toward Covid-19, I want to focus the new topic on Covid-19 and try to discover the difference toward Covid-19 between the two cultures. I also considered working on the comments since comments is a broader view of how people think other than the news channel and the comments has less constraint on the content and formation. Based on the comments, we can also see the attitude difference toward an event between the two culture. I will start will collecting the new dataset, bring out the analysis on the dataset, and then try to work on the comment of videos.

Collecting data

US Video

I started from collecting the US videos. The main resources I used is YouTube since it is the most popular video platform in the US. It is also the platform where people can upload their own videos so that there would be less constraint on the video content and video uploader. Since YouTube is the most used platform, there are existing APIs and websites that can do the downloading and related work, so that it would be much easier to do our work.

The first usable package in Python is the pytube [https://github.com/pytube/pytube]. You can follow the instruction on their Github site to download the video, one example can be below:



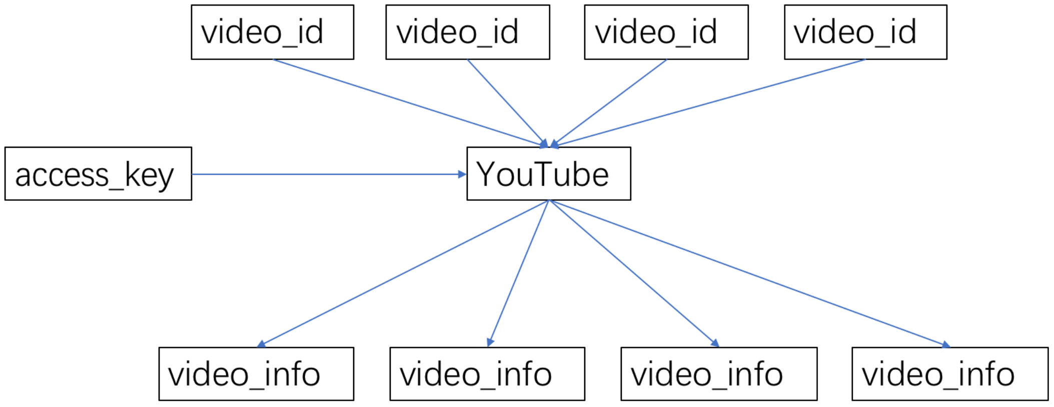
```python
from pytube import YouTube
yt = YouTube('http://youtube.com/watch?v=2lAe1cqCOXo')
yt.streams.filter(progressive=True, file_extension='mp4').order_by('resolution').desc().first().download()
```

```
'/content/YouTube Rewind 2019 For the Record  YouTubeRewind.mp4'
```

The advantage of pytube is that you can easily download the original video, whereas the disadvantages of the pytube are that you cannot get other information such as the channel_name and you cannot get the caption neither. An additional note is that when I first try pytube, it does not work and it kept reporting errors. After one about one month, it worked. The reason is that large companies such as YouTube, are periodically changing their website and their html file to prohibit others from scraping their information. Each time when the website change, it will take some time for the team to figure out the solution to use the API. This will also be one of the disadvantages of pytube.

Other than pytube, there are some websites that can help downloading the videos if you input the link of the video. I used this method because when I tried to download the videos, pytube did not work out. There is the website I used for Youtube Downloader [https://yt1s.com/en5].

Another technique I used is the official YouTube API. You will need to obtain an API key to use the official API. With that API key, you can create a query and send that query to the backend of the YouTube, then they will send back a response of the information you want.

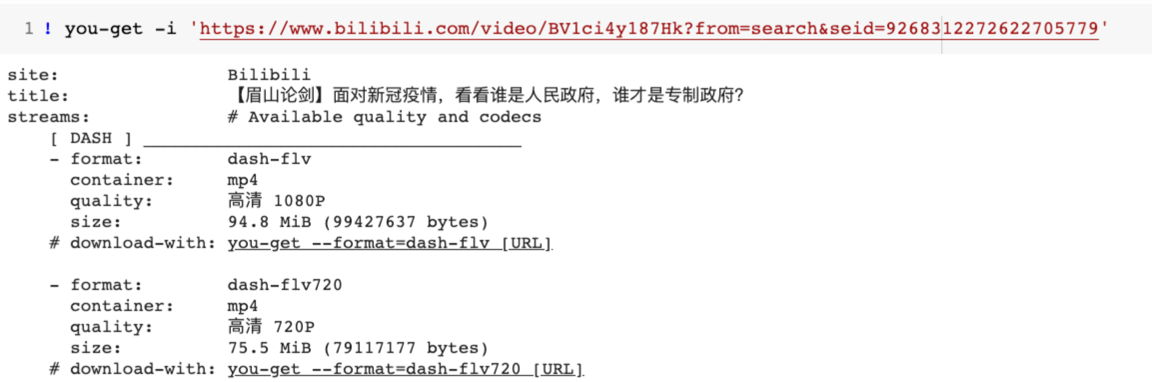


The advantage is that you will be able to get all the information about the video such as the video_id, channel_name and descriptions. After getting the descriptions of the videos, I realized that I still have the advertisement problems as students in previous semesters. There are some subscription links and the links of their official news website in the full description. By observations, I notice that the majority of the videos have their true descriptions in the first paragraph and the remaining parts are the ads, so I only kept the first paragraph of the video descriptions in the final dataset. Even though you can get plenty of information about the videos, the YouTube API does not allow uses to get access to their metadata. By saying this, it means that you cannot download the videos or the captions from the YouTube API.

Since I want to mostly work on the content of the video, especially what the video says, so I need the captions of the videos I collected. There are two types of captions that YouTube provides, one is the caption uploaded by the uploader, the other one is the auto-generated captions that were generated by YouTube. The caption uploaded by user will be more accurate but I also looked at the auto-generated captions which also have a very high accuracy. I used another website to download the captions which is this link: https://downsub.com/. By providing the video link, you can get access to all version of captions. There are also two types of the caption file you can get, one is the src file which should be having the time and corresponding captions, the other one is simply txt file. Since I am not creating the caption, but I only need the content of the caption, I only downloaded the txt file.

Chinese Video

For the Chinese videos, there is not a dominant video platform for Chinese videos. Videos are broke up by companies such as bilibili, iQiyi and youku. Among all of them, bilibili has the highest similarity with YouTube since it also allows any user to upload their own videos so it is a combination of videos uploaded by formal news companies and individual users. Since there is not a dominant platform, there is no official API for us to download the videos and information of videos. There is a python package called you-get [https://github.com/soimort/you-get], which can help downloading videos given the link.



```
! you-get -i 'https://www.bilibili.com/video/BV1ci4y187Hk?from=search&seid=9268312272622705779'
```

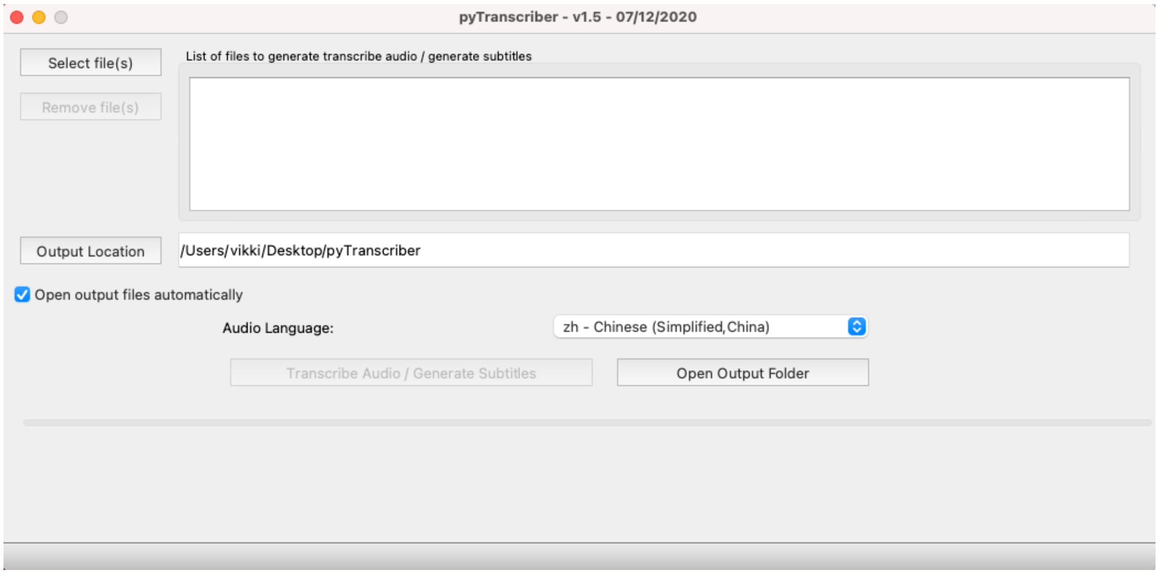
```
site:                Bilibili
title:               【眉山论剑】面对新冠疫情，看看谁是人民政府，谁才是专制政府?
streams:             # Available quality and codecs
    [ DASH ] ____________________________________
    - format:        dash-flv
      container:     mp4
      quality:       高清 1080P
      size:          94.8 MiB (99427637 bytes)
    # download-with: you-get --format=dash-flv [URL]

    - format:        dash-flv720
      container:     mp4
      quality:       高清 720P
      size:          75.5 MiB (79117177 bytes)
    # download-with: you-get --format=dash-flv720 [URL]
```

It can download multiple file types and video qualities. I realized that it will download videos and sound of the videos separately if I select the mp4 version, so I downloaded the flv360 version for all videos. This ensures the dataset to be small enough to be used in the future, but it also brings the problem that we need to convert the flv file type to mp4 type.

Next I found a website [https://cloudconvert.com/flv-to-mp4] which can help converting the flv file to mp4, but it has a constraints on the length of video you can convert every day.

Next step is that we also need to obtain the captions of Chinese videos. Since bilibi has recently added the subtitle function, but most videos does not have the separated subtitle, so we need to translate them using other apps. There is an app called pyTranscriber, you can download the app from their Github site: https://github.com/raryelcostasouza/pyTranscriber.


pyTranscriber - v1.5 - 07/12/2020
Select file(s)
Remove file(s)
List of files to generate transcribe audio / generate subtitles
Output Location
Transcribe Audio / Generate Subtitles
Open Output Folder

Output Location /Users/vikki/Desktop/pyTranscriber

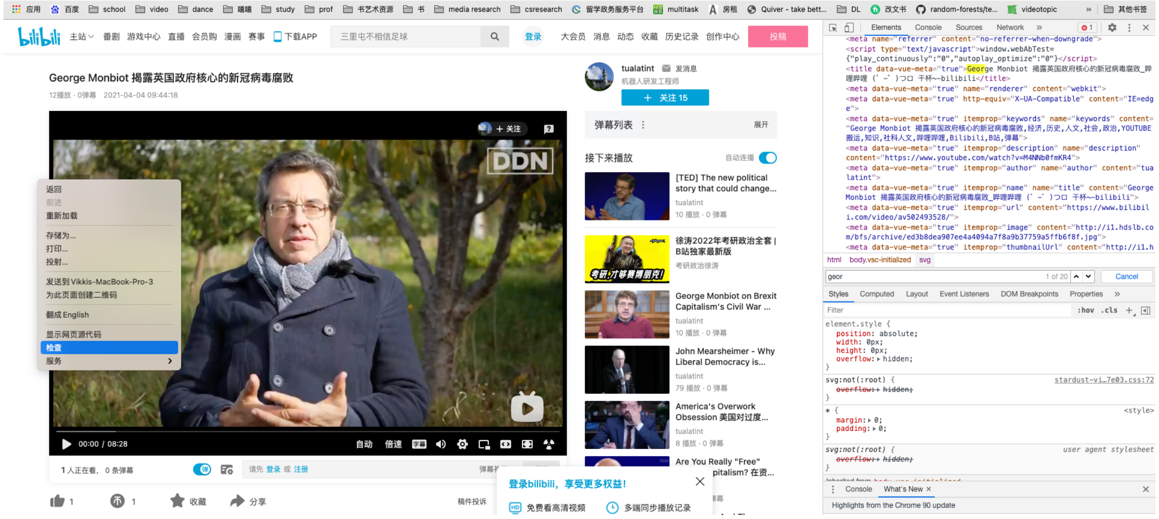
☑ Open output files automatically

Audio Language: zh - Chinese (Simplified,China)

Users are allowed to choose any type of language the video is, but it only accepts the videos in mp4, that is why I needed the previous step to convert flv to mp4.

Beyond that, we also want to get the information about videos such as the channel and description. Since we do not have the official API, we need the html scrapping to get the title, channel, and description. I used the a python package called BeautifulSoup [https://www.crummy.com/software/BeautifulSoup/bs4/doc/]. It is a python library for pulling data out HTML and XML file. This package can help neatly parse the HTML into readable formation and extract the main information you need.

Here is an example:



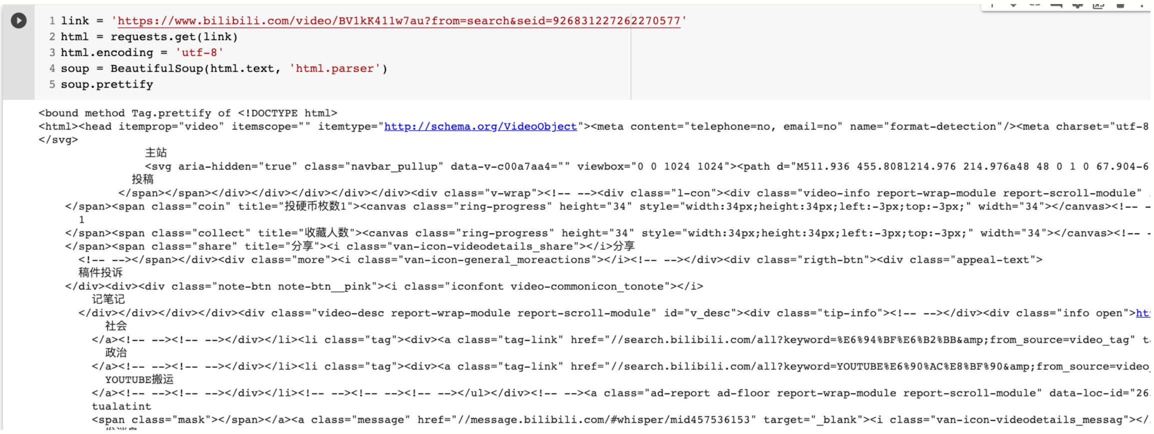
George Monbiot 揭露英国政府核心的新冠病毒腐败

12播放 · 0弹幕 2021-04-04 09:44:18

```
<meta name="referrer" content="no-referrer-when-downgrade">
<script type="text/javascript">window.webAbTest=
{"play_continuously":"0","autoplay_optimize":"0"}</script>
<title data-vue-meta="true">George Monbiot 揭露英国政府核心的新冠病毒腐败_哔
哩哔哩 (゜-゜)つロ 干杯~-bilibili</title>
<meta data-vue-meta="true" name="renderer" content="webkit">
<meta data-vue-meta="true" http-equiv="X-UA-Compatible" content="IE=edg
e">
<meta data-vue-meta="true" itemprop="keywords" name="keywords" content=
"George Monbiot 揭露英国政府核心的新冠病毒腐败,经济,历史,人文,社会,政治,YOUTUBE
搬运,知识,社科人文,哔哩哔哩,Bilibili,B站,弹幕">
<meta data-vue-meta="true" itemprop="description" name="description"
content="https://www.youtube.com/watch?v=M4NNb0fmKR4">
<meta data-vue-meta="true" itemprop="author" name="author" content="tua
latint">
<meta data-vue-meta="true" itemprop="name" name="title" content="George
Monbiot 揭露英国政府核心的新冠病毒腐败_哔哩哔哩 (゜-゜)つロ 干杯~-bilibili">
<meta data-vue-meta="true" itemprop="url" content="https://www.bilibil
i.com/video/av502493528/">
<meta data-vue-meta="true" itemprop="image" content="http://i1.hdslb.co
m/bfs/archive/ed3b8dea907ee4a4094a7f8a9b37759a5ffb6f8f.jpg">
<meta data-vue-meta="true" itemprop="thumbnailUrl" content="http://i1.h
```

```
element.style {
  position: absolute;
  width: 0px;
  height: 0px;
  overflow: ▸ hidden;
}
```

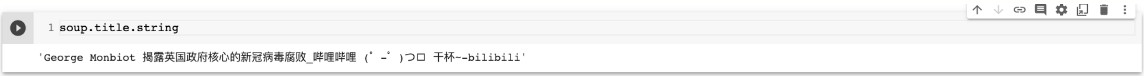
Given a page in bilibili, right click the page and choose the ‘check’, you will be able to see the original html file which makes up the web page. By taking a look, the titles are all contained in the html file. However we need to get this file in python.



```
link = 'https://www.bilibili.com/video/BV1kK411w7au?from=search&seid=926831227262270577'
html = requests.get(link)
html.encoding = 'utf-8'
soup = BeautifulSoup(html.text, 'html.parser')
soup.prettify
```

```
<bound method Tag.prettify of <!DOCTYPE html>
<html><head itemprop="video" itemscope="" itemtype="http://schema.org/VideoObject"><meta content="telephone=no, email=no" name="format-detection"/><meta charset="utf-8
</svg>
                主站
                <svg aria-hidden="true" class="navbar_pullup" data-v-c00a7aa4="" viewbox="0 0 1024 1024"><path d="M511.936 455.8081214.976 214.976a48 48 0 1 0 67.904-6
              投稿
            </span></span></div></div></div></div></div><div class="v-wrap"><!-- --><div class="l-con"><div class="video-info report-wrap-module report-scroll-module"
    </span><span class="coin" title="投硬币枚数1"><canvas class="ring-progress" height="34" style="width:34px;height:34px;left:-3px;top:-3px;" width="34"></canvas><!-- -
      1
    </span><span class="collect" title="收藏人数"><canvas class="ring-progress" height="34" style="width:34px;height:34px;left:-3px;top:-3px;" width="34"></canvas><!-- -
    </span><span class="share" title="分享"><i class="van-icon-videodetails_share"></i>分享
      <!-- --></span></div><div class="more"><i class="van-icon-general_moreactions"></i><!-- --></div><div class="rigth-btn"><div class="appeal-text">
      稿件投诉
    </div><div><div class="note-btn note-btn__pink"><i class="iconfont video-commonicon_tonote"></i>
        记笔记
      </div></div></div></div><div class="video-desc report-wrap-module report-scroll-module" id="v_desc"><div class="tip-info"><!-- --></div><div class="info open">ht
          社会
        </a><!-- --><!-- --></div></li><li class="tag"><div><a class="tag-link" href="//search.bilibili.com/all?keyword=%E6%94%BF%E6%B2%BB&amp;from_source=video_tag" t
          政治
        </a><!-- --><!-- --></div></li><li class="tag"><div><a class="tag-link" href="//search.bilibili.com/all?keyword=YOUTUBE%E6%90%AC%E8%BF%90&amp;from_source=video
          YOUTUBE搬运
        </a><!-- --><!-- --></div></li><!-- --><!-- --><!-- --></ul></div><!-- --><a class="ad-report ad-floor report-wrap-module report-scroll-module" data-loc-id="26
        tualatint
        <span class="mask"></span></a><a class="message" href="//message.bilibili.com/#whisper/mid457536153" target="_blank"><i class="van-icon-videodetails_messag"></
```

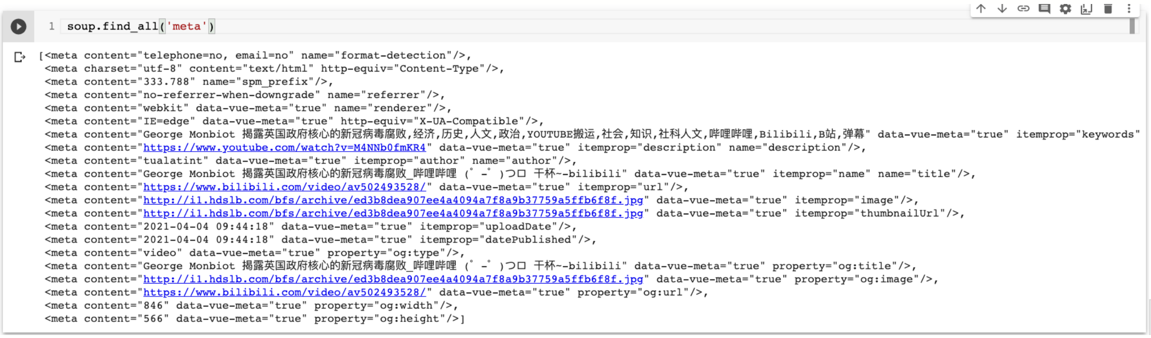
By sending a request to the link I got the html file, then I used BeautifulSoup to parse this html file and we can see that the new file are in the readable format. Note that the html.encoding=’ utf-8’ ensures that all Chinese characters can be printed out, otherwise all Chinese characters will be replaced by garbled characters.



```
soup.title.string
```

```
'George Monbiot 揭露英国政府核心的新冠病毒腐败_哔哩哔哩 (゜-゜)つロ 干杯~-bilibili'
```

Noticing that the titles always have the bilibili suffix, so we separate the string by ‘_’ .



```
soup.find_all('meta')
```

```
[<meta content="telephone=no, email=no" name="format-detection"/>,
 <meta charset="utf-8" content="text/html" http-equiv="Content-Type"/>,
 <meta content="333.788" name="spm_prefix"/>,
 <meta content="no-referrer-when-downgrade" name="referrer"/>,
 <meta content="webkit" data-vue-meta="true" name="renderer"/>,
 <meta content="IE=edge" data-vue-meta="true" http-equiv="X-UA-Compatible"/>,
 <meta content="George Monbiot 揭露英国政府核心的新冠病毒腐败,经济,历史,人文,政治,YOUTUBE搬运,社会,知识,社科人文,哔哩哔哩,Bilibili,B站,弹幕" data-vue-meta="true" itemprop="keywords"
 <meta content="https://www.youtube.com/watch?v=M4NNb0fmKR4" data-vue-meta="true" itemprop="description" name="description"/>,
 <meta content="tualatint" data-vue-meta="true" itemprop="author" name="author"/>,
 <meta content="George Monbiot 揭露英国政府核心的新冠病毒腐败_哔哩哔哩 (゜-゜)つロ 干杯~-bilibili" data-vue-meta="true" itemprop="name" name="title"/>,
 <meta content="https://www.bilibili.com/video/av502493528/" data-vue-meta="true" itemprop="url"/>,
 <meta content="http://i1.hdslb.com/bfs/archive/ed3b8dea907ee4a4094a7f8a9b37759a5ffb6f8f.jpg" data-vue-meta="true" itemprop="image"/>,
 <meta content="http://i1.hdslb.com/bfs/archive/ed3b8dea907ee4a4094a7f8a9b37759a5ffb6f8f.jpg" data-vue-meta="true" itemprop="thumbnailUrl"/>,
 <meta content="2021-04-04 09:44:18" data-vue-meta="true" itemprop="uploadDate"/>,
 <meta content="2021-04-04 09:44:18" data-vue-meta="true" itemprop="datePublished"/>,
 <meta content="video" data-vue-meta="true" property="og:type"/>,
 <meta content="George Monbiot 揭露英国政府核心的新冠病毒腐败_哔哩哔哩 (゜-゜)つロ 干杯~-bilibili" data-vue-meta="true" property="og:title"/>,
 <meta content="http://i1.hdslb.com/bfs/archive/ed3b8dea907ee4a4094a7f8a9b37759a5ffb6f8f.jpg" data-vue-meta="true" property="og:image"/>,
 <meta content="https://www.bilibili.com/video/av502493528/" data-vue-meta="true" property="og:url"/>,
 <meta content="846" data-vue-meta="true" property="og:width"/>,
 <meta content="566" data-vue-meta="true" property="og:height"/>]
```

By taking a look at the meta data, we notice that the eighth line is the description and next line is author, so we also extract those information and put that into the csv.

About the final Dataset

The final dataset is on Google Drive and can be access here: https://drive.google.com/drive/folders/1VKwzykZHrOT4Zr-ikiPDKbXAba2MfxSC?usp=sharing

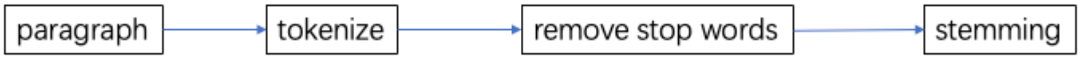
When I made the query on YouTube and bilibili, I realized that since Covid-19 has been in our life for too long time, the topic also spread out a lot. If I only search for Covid-19, the videos coming out will be in different sub-topic. Based on this fact, I decided to break out the videos to be three part, the first sub-topic is ‘covid-19 federal’ , the second topic is ‘covid-19 vaccination’ , and the third topic is ‘covid-19 trump’ . Each topic consists 15 videos from YouTube and 15 videos from bilibili. I chose these three topics because I can see the culture difference in those topics. For the federal topic, the US government is announcing much less severity of the condition than the Chinese government. For the vaccination, the condition in US is that there are more people who want to get vaccinated than the available vaccination, but the condition in China is that people are avoiding to get vaccinated. For the Trump topic, when president Trump tested positive for Covid-19, I think different parties have different attitude toward this event in the US but the attitude in China is more unified. I hope to find out some difference through these three sub-topics.

The English captions can be directly downloaded, but after I translated all Chinese videos, there is one problem. For some Chinese videos, the language they used are not only Chinese, but a combination of English and some other language. When they refer to new videos in another language, they would use the original audio and include the translate in the scene. There are also some video that the audio itself is a piece of music and all news contents are printed in the scene. For those videos, the translate is too inaccurate to reflect the news video. Because of this difficulty, my analysis was mainly concentrated on US videos for this semester.

Caption Processing: tf-idf

NLTK Packages

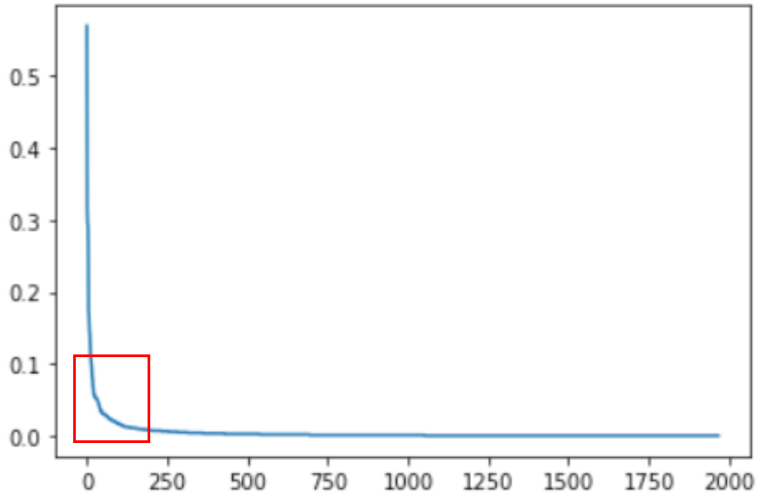
I used the NLTK packages to do the first three step of the pre-processing.



After getting a paragraph, I first tokenized the paragraph using the existing function in NLTK. NLTK also contains a dictionary of all stop words in English, I remove the stop words in the list of tokens. Next, since some word has many forms, I did the stemming to the word list also using NLTK.

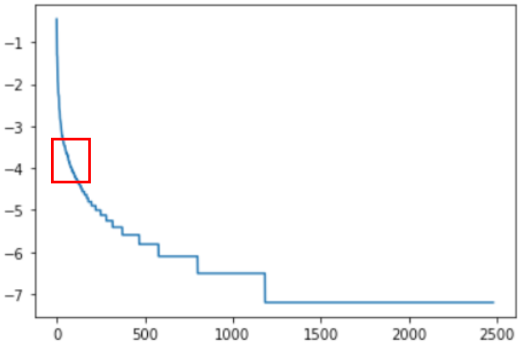
Next step is that calculating the TF-IDF of the word list. TF is the term frequency of a word in a document, and IDF is the inverse document frequency of the word across a set of documents, it is a measure of how much information the word provides.

Take the federal topic as an example, if we plot the IF-IDF plot:

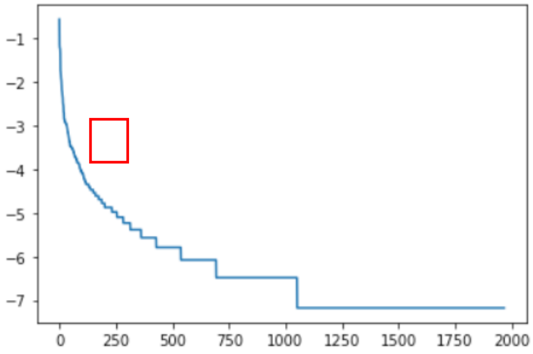


We notice that we only interested in the red area because words with too high tfidf tends to words that do not provide much meanings such as the, that, not, how, and words with too low tfidf usually rarely appears in the documents. With the sharp decrease, it is hard to tell which words we want, so I took the logarithm of tfidf.

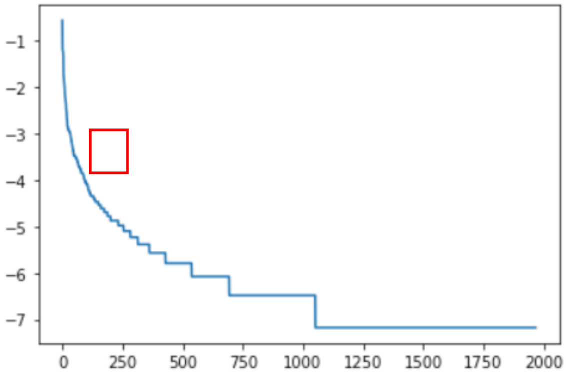
For federal, the log(tf-idf) is:



For vaccination, the log(tf-idf) is:



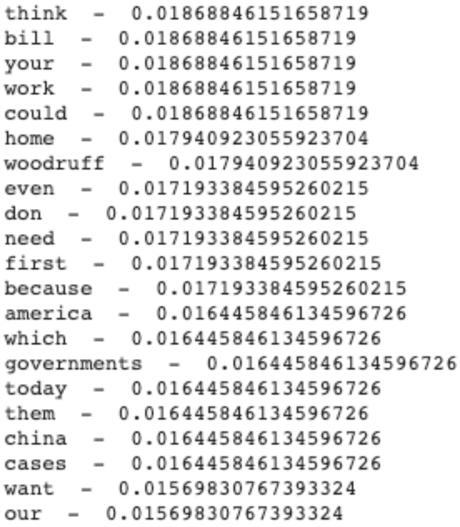
For Trump, the log(tf-idf) is:



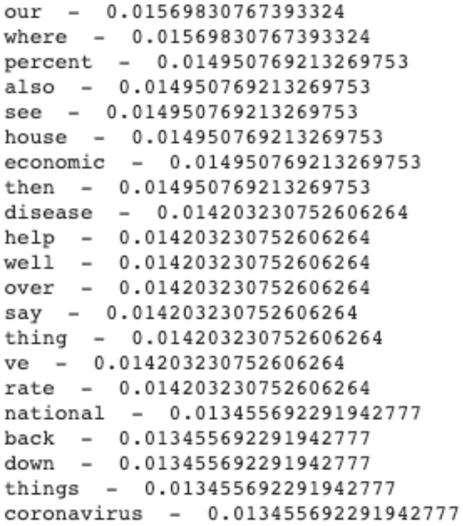
We are more interested in the red area where the tf-idf just starts to decrease. Because those words usually have some meaning and also appears relatively often in the documents.

By shorting the word-log(tfidf) dictionary and finding the relative matching number, there is a list displaying those words:

For the federal topic:



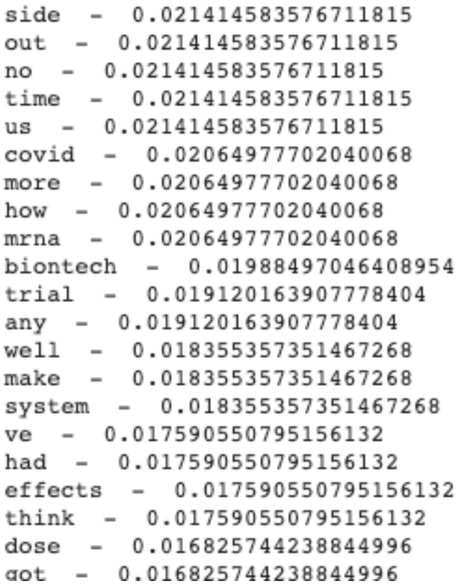
```
think  -  0.01868846151658719
bill  -  0.01868846151658719
your  -  0.01868846151658719
work  -  0.01868846151658719
could  -  0.01868846151658719
home  -  0.017940923055923704
woodruff  -  0.017940923055923704
even  -  0.017193384595260215
don  -  0.017193384595260215
need  -  0.017193384595260215
first  -  0.017193384595260215
because  -  0.017193384595260215
america  -  0.016445846134596726
which  -  0.016445846134596726
governments  -  0.016445846134596726
today  -  0.016445846134596726
them  -  0.016445846134596726
china  -  0.016445846134596726
cases  -  0.016445846134596726
want  -  0.01569830767393324
our  -  0.01569830767393324
```



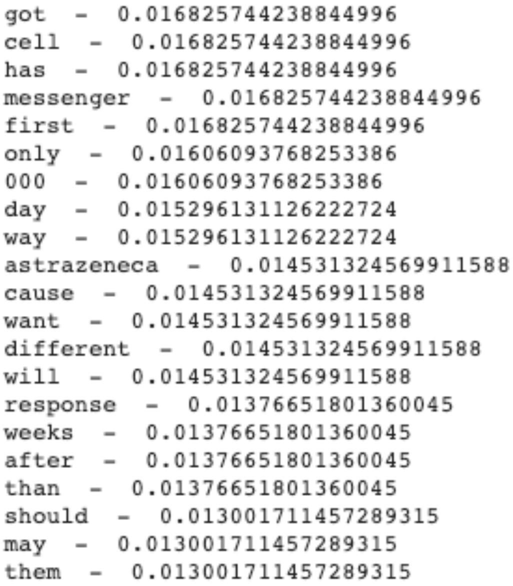
```
our  -  0.01569830767393324
where  -  0.01569830767393324
percent  -  0.014950769213269753
also  -  0.014950769213269753
see  -  0.014950769213269753
house  -  0.014950769213269753
economic  -  0.014950769213269753
then  -  0.014950769213269753
disease  -  0.014203230752606264
help  -  0.014203230752606264
well  -  0.014203230752606264
over  -  0.014203230752606264
say  -  0.014203230752606264
thing  -  0.014203230752606264
ve  -  0.014203230752606264
rate  -  0.014203230752606264
national  -  0.013455692291942777
back  -  0.013455692291942777
down  -  0.013455692291942777
things  -  0.013455692291942777
coronavirus  -  0.013455692291942777
```

We notice that there are words like bill, government, china, national, which are more country-wise.

For the vaccination topic:



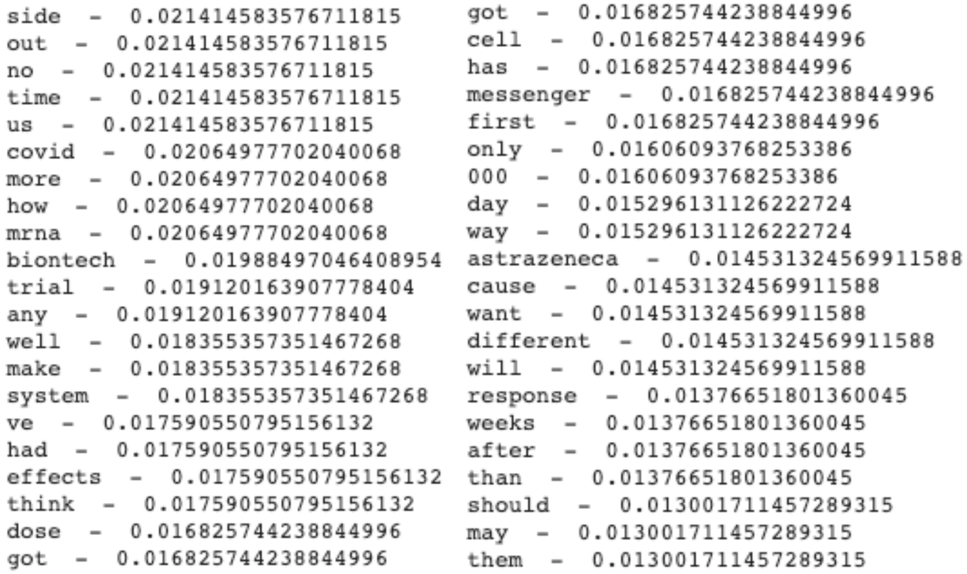
```
side  -  0.021414583576711815
out  -  0.021414583576711815
no  -  0.021414583576711815
time  -  0.021414583576711815
us  -  0.021414583576711815
covid  -  0.02064977702040068
more  -  0.02064977702040068
how  -  0.02064977702040068
mrna  -  0.02064977702040068
biontech  -  0.01988497046408954
trial  -  0.019120163907778404
any  -  0.019120163907778404
well  -  0.018355357351467268
make  -  0.018355357351467268
system  -  0.018355357351467268
ve  -  0.017590550795156132
had  -  0.017590550795156132
effects  -  0.017590550795156132
think  -  0.017590550795156132
dose  -  0.016825744238844996
got  -  0.016825744238844996
```



```
got  -  0.016825744238844996
cell  -  0.016825744238844996
has  -  0.016825744238844996
messenger  -  0.016825744238844996
first  -  0.016825744238844996
only  -  0.01606093768253386
000  -  0.01606093768253386
day  -  0.015296131126222724
way  -  0.015296131126222724
astrazeneca  -  0.014531324569911588
cause  -  0.014531324569911588
want  -  0.014531324569911588
different  -  0.014531324569911588
will  -  0.014531324569911588
response  -  0.01376651801360045
weeks  -  0.01376651801360045
after  -  0.01376651801360045
than  -  0.01376651801360045
should  -  0.013001711457289315
may  -  0.013001711457289315
them  -  0.013001711457289315
```

We notice that there are words like effects, system, dose, biontech which are more biology type.

For the Trump topic:



```
side  -  0.021414583576711815
out  -  0.021414583576711815
no  -  0.021414583576711815
time  -  0.021414583576711815
us  -  0.021414583576711815
covid  -  0.02064977702040068
more  -  0.02064977702040068
how  -  0.02064977702040068
mrna  -  0.02064977702040068
biontech  -  0.01988497046408954
trial  -  0.019120163907778404
any  -  0.019120163907778404
well  -  0.018355357351467268
make  -  0.018355357351467268
system  -  0.018355357351467268
ve  -  0.017590550795156132
had  -  0.017590550795156132
effects  -  0.017590550795156132
think  -  0.017590550795156132
dose  -  0.016825744238844996
got  -  0.016825744238844996
```

```
got  -  0.016825744238844996
cell  -  0.016825744238844996
has  -  0.016825744238844996
messenger  -  0.016825744238844996
first  -  0.016825744238844996
only  -  0.0160609376825338 6
000  -  0.01606093768253386
day  -  0.015296131126222724
way  -  0.015296131126222724
astrazeneca  -  0.014531324569911588
cause  -  0.014531324569911588
want  -  0.014531324569911588
different  -  0.014531324569911588
will  -  0.014531324569911588
response  -  0.01376651801360045
weeks  -  0.01376651801360045
after  -  0.01376651801360045
than  -  0.01376651801360045
should  -  0.013001711457289315
may  -  0.013001711457289315
them  -  0.013001711457289315
```

The Trump topic does not have very distinguishable words, because when we query Trump on YouTube, the videos are more of a combination of many related topics instead of concentrating on the topic "Trump tested positive".

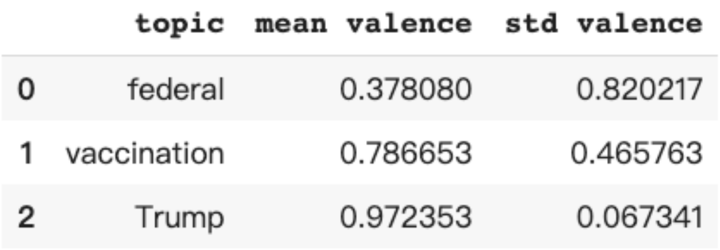
We can see that even though we have already changed the tf-idf range, there are still many words in the list which does not have 'real meaning' in the context.

Arousal and Valence Analysis

The next part I worked on is the sentiment analysis of news, since we have already got the captions of news. Sentiment analysis is the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. I will do the analysis from two perspectives, valence and arousal. Valence, on the other hand, is the level of pleasantness that an event generates and is defined along a continuum from negative to positive. Arousal (or intensity) is the level of autonomic activation that an event creates, and ranges from calm (or low) to excited (or high).

Valence

There is a python package called Vader [https://github.com/cjhutto/vaderSentiment]. It is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. This model is trained on four data sets, a tweets data set, a Movie data set, a financial news data set, and a New York Times data set. For each sentence, there is a sentiment score representing how positive it is and how negative it is and the sentiment score will be ranged from -3 to 3. This score is rated by human rater, and those people were trained by them. Each sentence will be rated by three people and take the mean of their ratings. Since this sentiment rating score represent the positive and negative attitude in the sentence, I treated this score as the valence score. By loading the Vader package in python, it can provide four scores, the first one is the compound score which represent the overall valence, it also has a ‘neg’ score which is how negative the sentence is, a ‘neu’ score which is how neutral the sentence is, and a ‘pos’ score which is how positive the sentence is. By going over each text to calculate the valence score, below is the result:



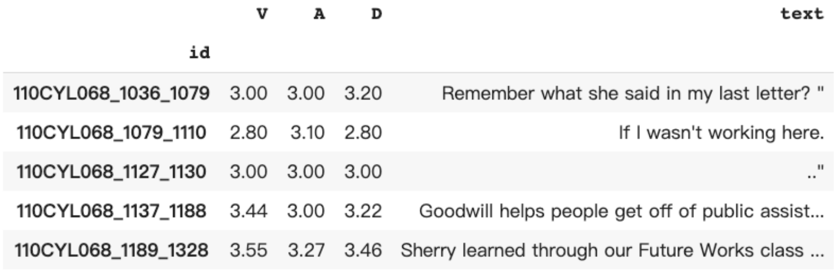
	topic	mean valence	std valence
0	federal	0.378080	0.820217
1	vaccination	0.786653	0.465763
2	Trump	0.972353	0.067341

We can see that even those all three topics have positive mean valence score, federal topic has the lowest valence score, and it also has the highest variance, this means that the federal topic has the least positive attitude, and among all videos, the attitude is changing a lot. Whereas the topic of Trump is having the highest valence score and the lowest variance, it means the Trump topic is receiving the highest positive attitude and the attitude is relatively consistent among all videos. This result is quite surprising since I assumed the Trump topic would receive the lowest valence score and high variance. I looked back the data set to try to find the reason, and one possible reason is that when I queried the Trump topic on YouTube, most videos coming up are not Trump been tested positive. Instead, most videos are all about some policies that

Trump announced, or the country’ s condition, or the talk gave by Trump. The content probably caused the topic to get the highest valence score.

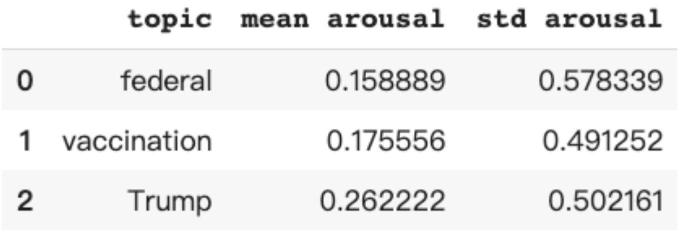
Arousal

For the arousal, I used an existing data set called EmoBank, it is a large-scale text corpus manually annotated with emotion according to the psychological Valence-Arousal-Dominance scheme. An example would be below:



	V	A	D	text
id				
110CYL068_1036_1079	3.00	3.00	3.20	Remember what she said in my last letter? "
110CYL068_1079_1110	2.80	3.10	2.80	If I wasn't working here.
110CYL068_1127_1130	3.00	3.00	3.00	.."
110CYL068_1137_1188	3.44	3.00	3.22	Goodwill helps people get off of public assist...
110CYL068_1189_1328	3.55	3.27	3.46	Sherry learned through our Future Works class ...

The column A is the arousal score of the text. Then I trained a NaiveBayes classifier using the training data set, then apply the model on the captions of the news videos and got the below result:

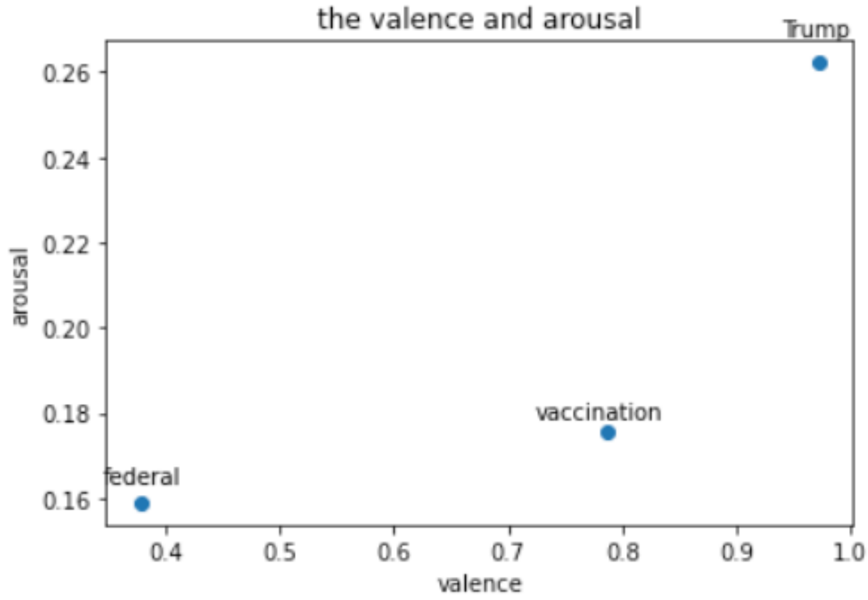


	topic	mean arousal	std arousal
0	federal	0.158889	0.578339
1	vaccination	0.175556	0.491252
2	Trump	0.262222	0.502161

From the form we can see that the federal topic and vaccination topic are having similar mean and variance, whereas Trump topic is having the highest arousal, and this result match my assumption.

Result

What I wanted to do was to create a plot with the x-axis to be valence and y-axis to be arousal. For the three topic and two languages, there should be six points in the plot. Right now, since I only have the English captions, I will only have three points on the plot now.



Future work

The main part of the future work would be to get the accurate Chinese captions, after get the Chinese caption, I will need to find a way to do the tf-idf analysis for the Chinese captions. For the English part, I mentioned that even though I have discarded the words with highest tf-idf, there are still many meaningless words in the range, so the future work would also include removing those meaningless word for the English caption. After I get the high tf-idf words for English and Chinese videos, I can start to work on the comparison.

The next part of the future work would be to do the valence and arousal analysis of the Chinese captions, and then put the three topics on the plot two. By comparing the different positions of the two points of same topic, we will be able to find out the cultural difference of the Covid-19 topic.

Reference:

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5597854/#:~:text=Arousal%20(or%20intensity)%20is%20the,continuum%20from%20negative%20to%20positive.

http://comp.social.gatech.edu/papers/icwsm14.vader.hutto.pdf

https://github.com/JULIELab/EmoBank