Cross Culture Analysis

Andy (sc4426@columbia.edu), Gary (xh2379@columbia.edu)

Introduction

Our main goal is to distinguish cultural differences in video news which focus on topics between multiple countries, let's say, China and United States, which have cultural backgrounds in contrast. The differences include the anchors' gestures, the text content of the news, the way they convey a perspective, the perspectives they have on same incident, etc.

In this project, the topic we mainly focus on is AlphaGo. The reason we think it is an appropriate topic to research on is because AlphaGo is a topic which people all around United States and China would show interests on. More specifically, AlphaGo, which is developed by Google, would definitely attract people's attention since the firm (Google) is a long-last red-hot company who possess the most advanced technologies. While the chess game "Go" is originally from China and it should be very popular in the place of its origin. To sum up, we think the epic duel between the Google developed AlphaGo and the genius Chinese player should attract enough attention for us to study.

Web Scraping

US Videos

We start our work by collecting data with web scraping. We mainly focus on the videos on Youtube for Alphago-related news from United States, since Youtube is currenlyt the most popular video platform. Another reason Youtube would be the best choice is that there exist plenty of APIs and SDKs which allow us to simplify the scraping works. However, it is another world in China since Youtube is not popular in the area. Hence, we separate the scraping task in aid of fitting to different cultural environments.

pytube

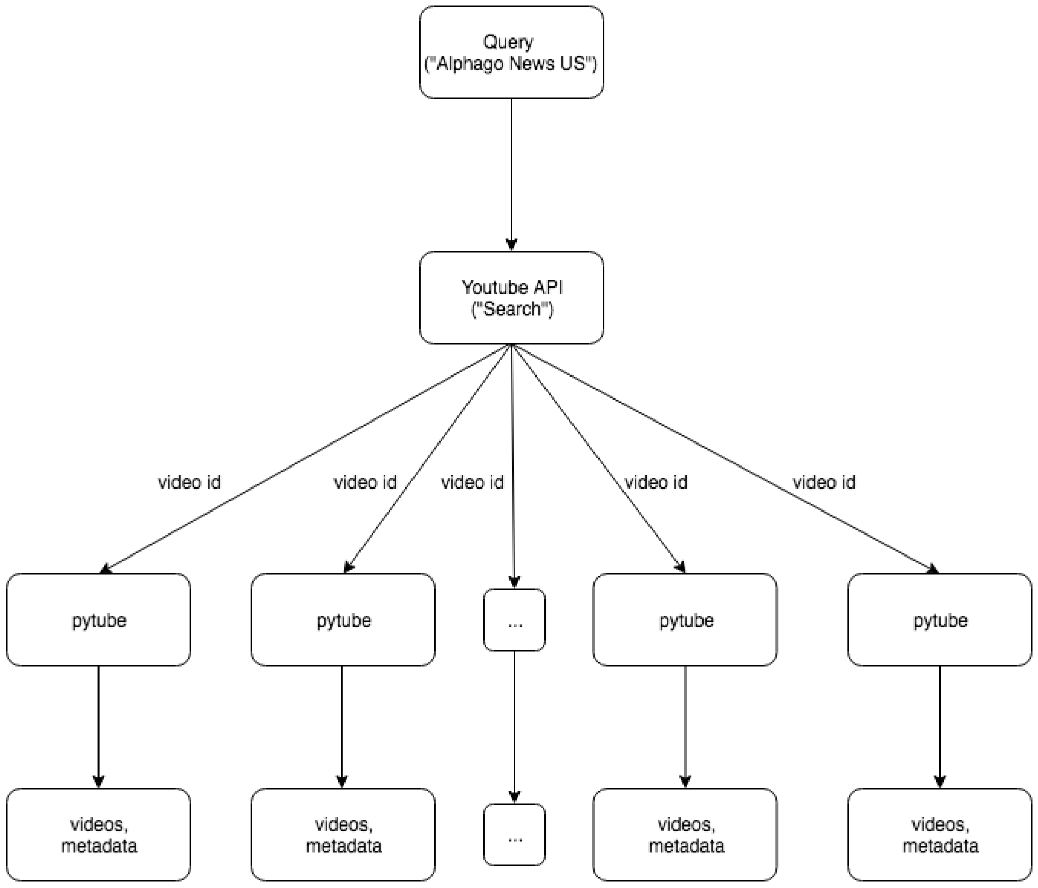
The whole scraping task is implemented with Python. We take advantage of the Python package called pytube to download videos from Youtube. The package supports not only videos, but extracting audio files and metadata such as descriptions, title, video length, etc. This package really simply our works since we do not have to crawl into the html on Youtube by ourselves with this package. However, Pytube can only be used in one specific video. In other words, we need to know which video we want to scrape first. To solve this problem, it is necessary for us to get the video IDs.

Youtube API

The official Youtube API provides services which we can implement to solve the problem mentioned above. There is a “Search” API which we can use it by sending requests, which include the query we want to search. In this project, our results are mainly generated from the query “Alphao News US”, in the way which we can focus on the video news from media companies from United States. The response contains list of video objects. The video objects contains ID, thumbnails, metadata, etc. The order of the list should be the same as we exactly search on Youtube’s website. In conclusion, the Youtube API also simplify our scraping works by providing a simple gateway to get the results we need in the search list page on Youtube (before the video page).

To sum up, we use the techniques mentioned above to do the scraping on Youtube videos. First, query "Alphago News US" with Youtube API and get the video IDs in the result. Then put the video IDs into pytube sdk to download the videos and metadata.

We download 500 videos with titles and queries from Youtube and extract the features from them. The methodologies of feature extracting will be discussed later on. We have not met any downloading limitation so far, which might happen in some cases when we are trying to send large amount of requests to other websites.



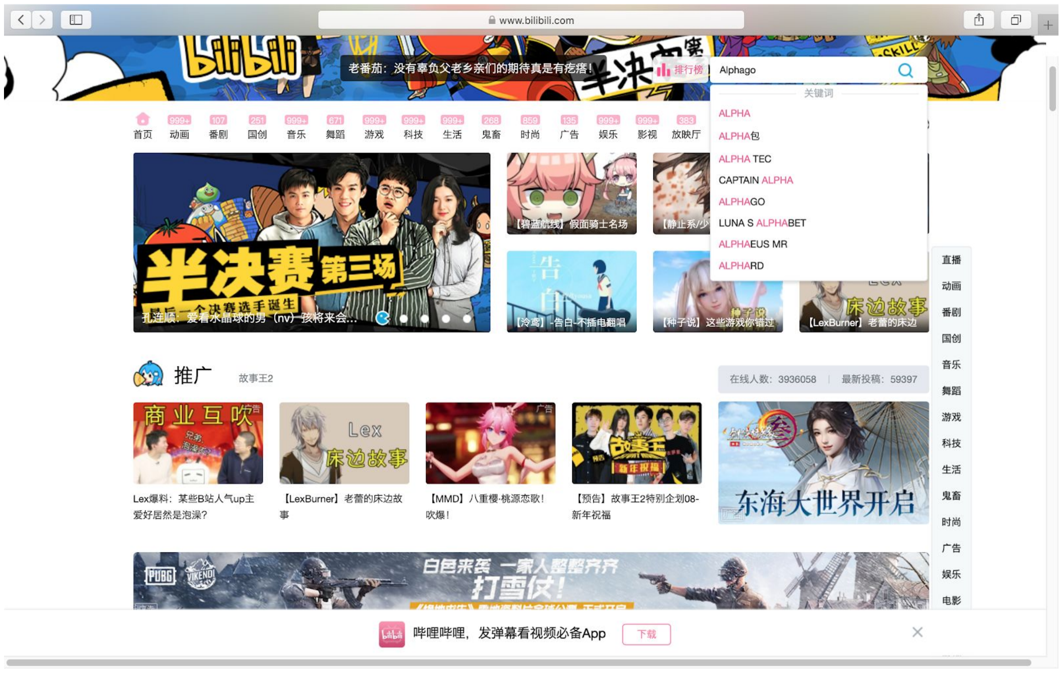
Chinese Videos

For Chinese videos, there is not a single dominating video source such as Youtube. Therefore, we decide to use several websites as our sources such as Bilibili (https://www.bilibili.com), Iqiyi (https://www.iqiyi.com) and Tencent Videos (https://v.qq.com). Unlike Youtube, which is very friendly to provide official APIs for easy downloading, the Chinese video websites mentioned above require more work to scrape and download. Details are discussed in the following sections.

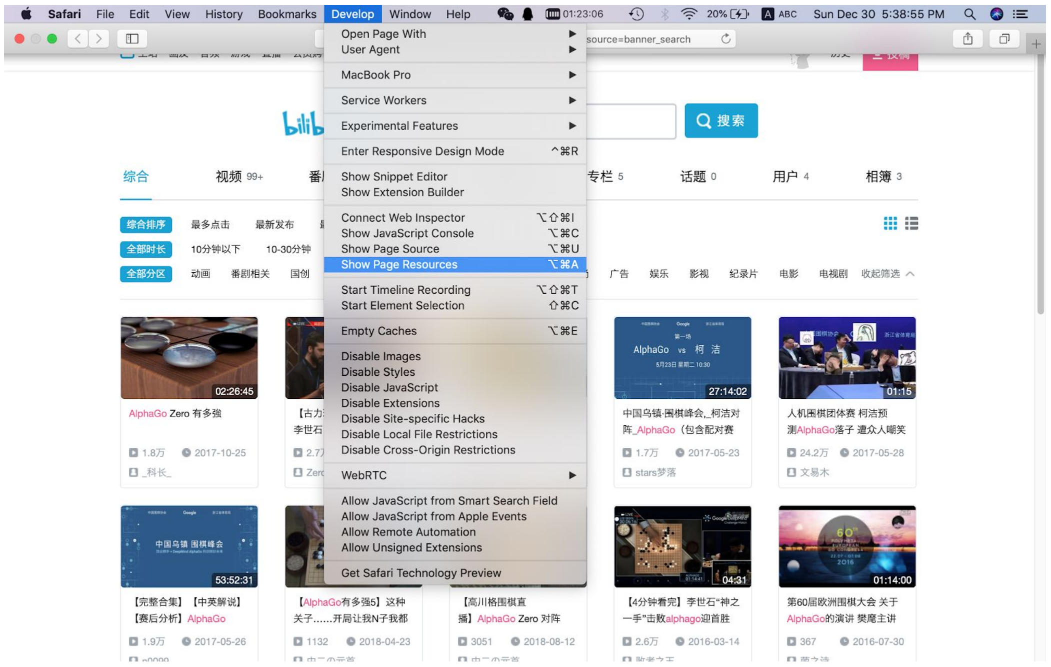
HTML Parsing – Obtaining URLs

All of the 3 websites mentioned provide a searching entry for users to input keywords. After making a query with some keywords, the first step for obtaining Chinese videos from those websites is to parse the HTML files returned by the sites in order to fetch the URLs for the videos. Following is a demonstration of doing so using the browser Safari.

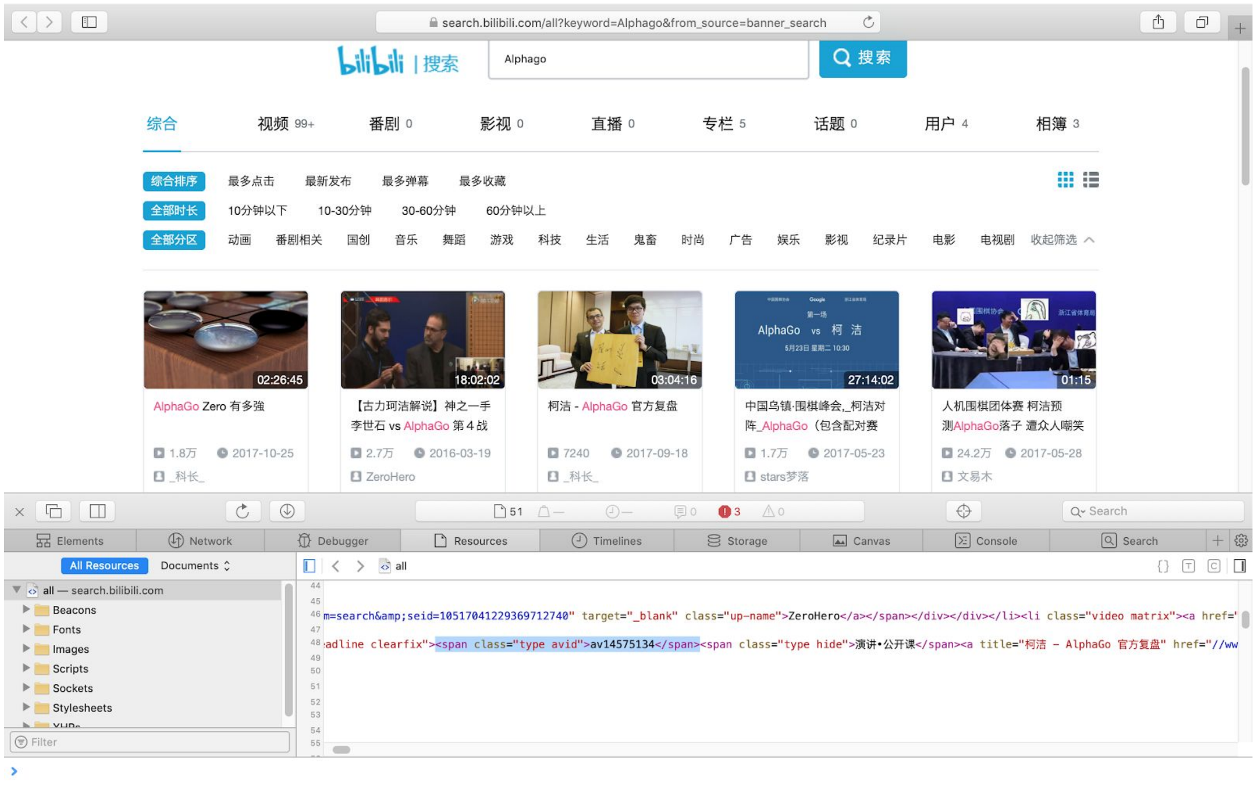
Step 1: Make a query for videos.



Step 2: After seeing the results, open the page source by choosing Develop in the menu bar followed by clicking Show Page Resources.

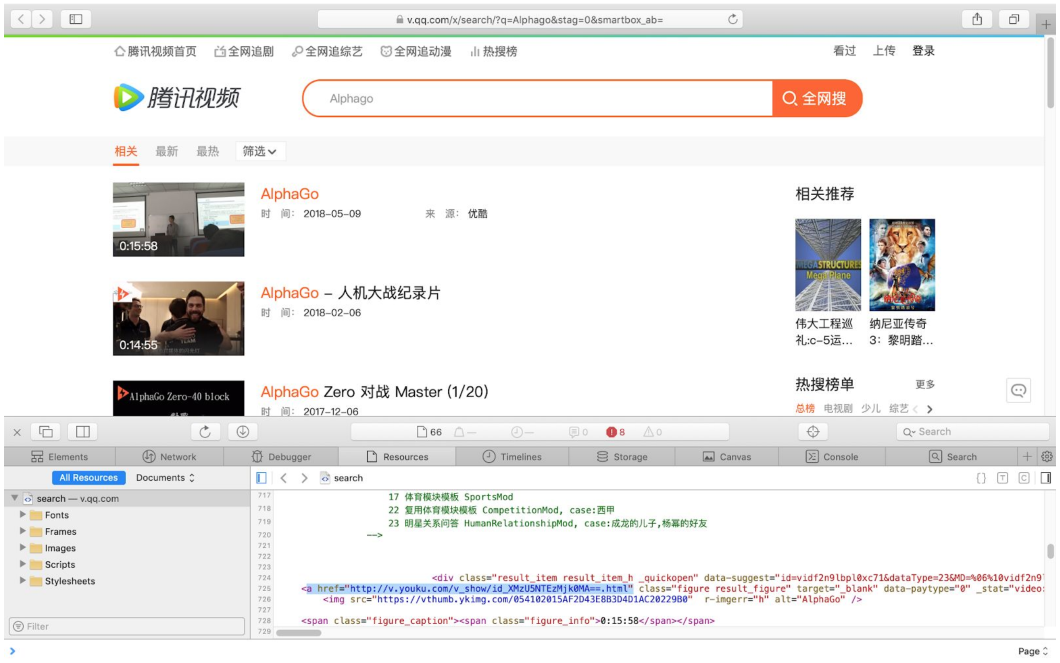

Safari
File
Edit
View
History
Bookmarks
Develop
Window
Help
Sun Dec 30 5:38:55 PM
Open Page With
User Agent
MacBook Pro
Service Workers
Experimental Features
Enter Responsive Design Mode ^⌘R
Show Snippet Editor
Show Extension Builder
Connect Web Inspector ⌥⇧⌘I
Show JavaScript Console ⌥⌘C
Show Page Source ⌥⌘U
Show Page Resources ⌥⌘A
Start Timeline Recording ⌥⇧⌘T
Start Element Selection ⇧⌘C
Empty Caches ⌥⌘E
Disable Images
Disable Styles
Disable JavaScript
Disable Extensions
Disable Site-specific Hacks
Disable Local File Restrictions
Disable Cross-Origin Restrictions
WebRTC
Allow JavaScript from Smart Search Field
Allow JavaScript from Apple Events
Allow Remote Automation
Allow Unsigned Extensions
Get Safari Technology Preview
搜索
综合
视频 99+
专栏 5
话题 0
用户 4
相簿 3
AlphaGo Zero 有多强
中国乌镇·围棋峰会_柯洁对阵_AlphaGo（包含配对赛
人机围棋团体赛 柯洁预测AlphaGo落子 遭众人嘲笑
【完整合集】【中英解说】【赛后分析】AlphaGo
【AlphaGo有多强5】这种关于……开局让我N子我都
【高川格围棋直播】AlphaGo Zero 对阵
【4分钟看完】李世石"神之一手"击败alphago迎首胜
第60届欧洲围棋大会 关于AlphaGo的演讲 樊麾主讲

Step 3: Different websites have different ways to store the URL links of the resulting videos, and in this step we need to inspect how the URLs can be obtained. Bilibili uses *avid* as a unique identifier for its videos and the URL simply consists of 'https://www.bilibili.com/video' plus the *avid*. We see that each video is wrapped in a <li class="video matrix"> tag and the *avid* is in a <span class="type avid"> tag.



```
m=search&amp;seid=10517041229369712740" target="_blank" class="up-name">ZeroHero</a></span></div></div></li><li class="video matrix"><a href="
adline clearfix"><span class="type avid">av14575134</span><span class="type hide">演讲•公开课</span><a title="柯洁 - AlphaGo 官方复盘" href="//ww
```

Tencent has its resulting videos' information stored in <div class="result_item result_item_h _quickopen" …> tags and the URL can be found in the <a href=...> tag inside.



AlphaGo
时 间: 2018-05-09 来 源: 优酷
0:15:58

AlphaGo – 人机大战纪录片
时 间: 2018-02-06
0:14:55

AlphaGo Zero 对战 Master (1/20)
时 间: 2017-12-06

伟大工程巡礼:c-5运… 纳尼亚传奇3: 黎明踏…

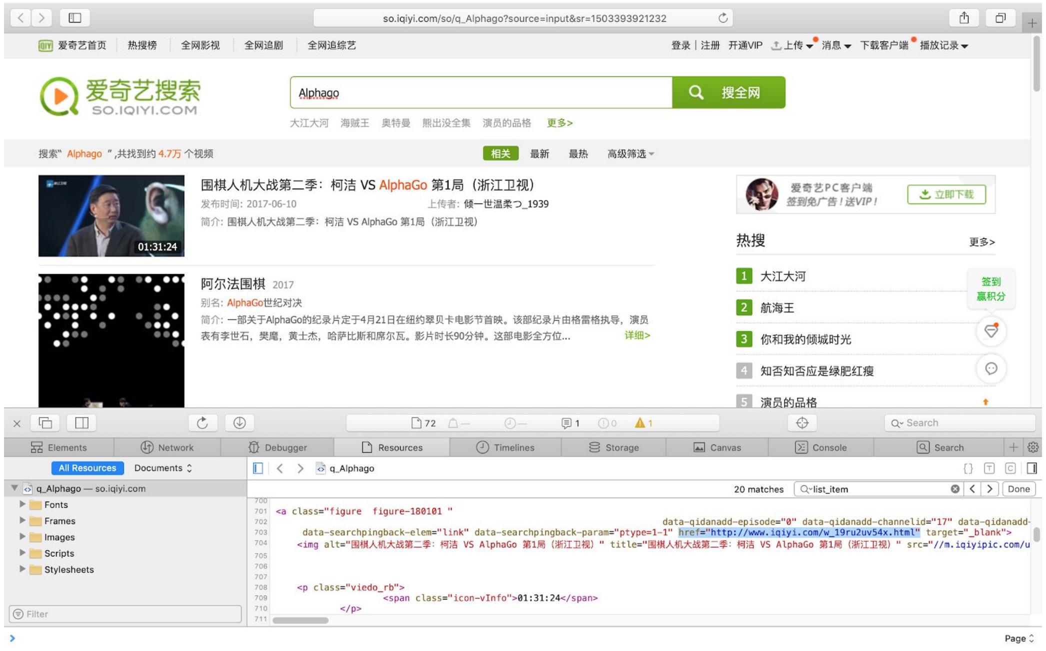
```
                   17 体育模块模板 SportsMod
                   22 复用体育模块模板 CompetitionMod, case:西甲
                   23 明星关系问答 HumanRelationshipMod, case:成龙的儿子,杨幂的好友
               -->



                        <div class="result_item result_item_h _quickopen" data-suggest="id=vidf2n9lbpl0xc71&dataType=23&MD=%06%10vidf2n9
   <a href="http://v.youku.com/v_show/id_XMzU5NTEzMjk0MA==.html" class="figure result_figure" target="_blank" data-paytype="0" _stat="video:
      <img src="https://vthumb.ykimg.com/054102015AF2D43E8B3D4D1AC20229B0"  r-imgerr="h" alt="AlphaGo" />

   <span class="figure_caption"><span class="figure_info">0:15:58</span></span>
```

Iqiyi stored the information in <li class="list_item" …> tags and the URL is in a <a … href=...> subtag.


so.iqiyi.com/so/q_Alphago?source=input&sr=1503393921232
爱奇艺首页 热搜榜 全网影视 全网追剧 全网追综艺
登录 | 注册 开通VIP 上传 消息 下载客户端 播放记录
爱奇艺搜索 SO.IQIYI.COM
Alphago
搜全网
大江大河 海贼王 奥特曼 熊出没全集 演员的品格 更多>
相关 最新 最热 高级筛选
爱奇艺PC客户端 签到免广告！送VIP！ 立即下载
热搜 更多>
签到赢积分
Elements Network Debugger Resources Timelines Storage Canvas Console Search
All Resources Documents
q_Alphago — so.iqiyi.com
Fonts
Frames
Images
Scripts
Stylesheets
Filter
20 matches
list_item
Done
Page

搜索" Alphago ",共找到约 4.7万 个视频

围棋人机大战第二季：柯洁 VS AlphaGo 第1局（浙江卫视）
发布时间: 2017-06-10 上传者: 倾一世温柔つ_1939
简介: 围棋人机大战第二季: 柯洁 VS AlphaGo 第1局（浙江卫视）
01:31:24

阿尔法围棋 2017
别名: AlphaGo世纪对决
简介: 一部关于AlphaGo的纪录片定于4月21日在纽约翠贝卡电影节首映。该纪录片由格雷格执导，演员表有李世石，樊麾，黄士杰，哈萨比斯和席尔瓦。影片时长90分钟。这部电影全方位... 详细>

1 大江大河
2 航海王
3 你和我的倾城时光
4 知否知否应是绿肥红瘦
5 演员的品格

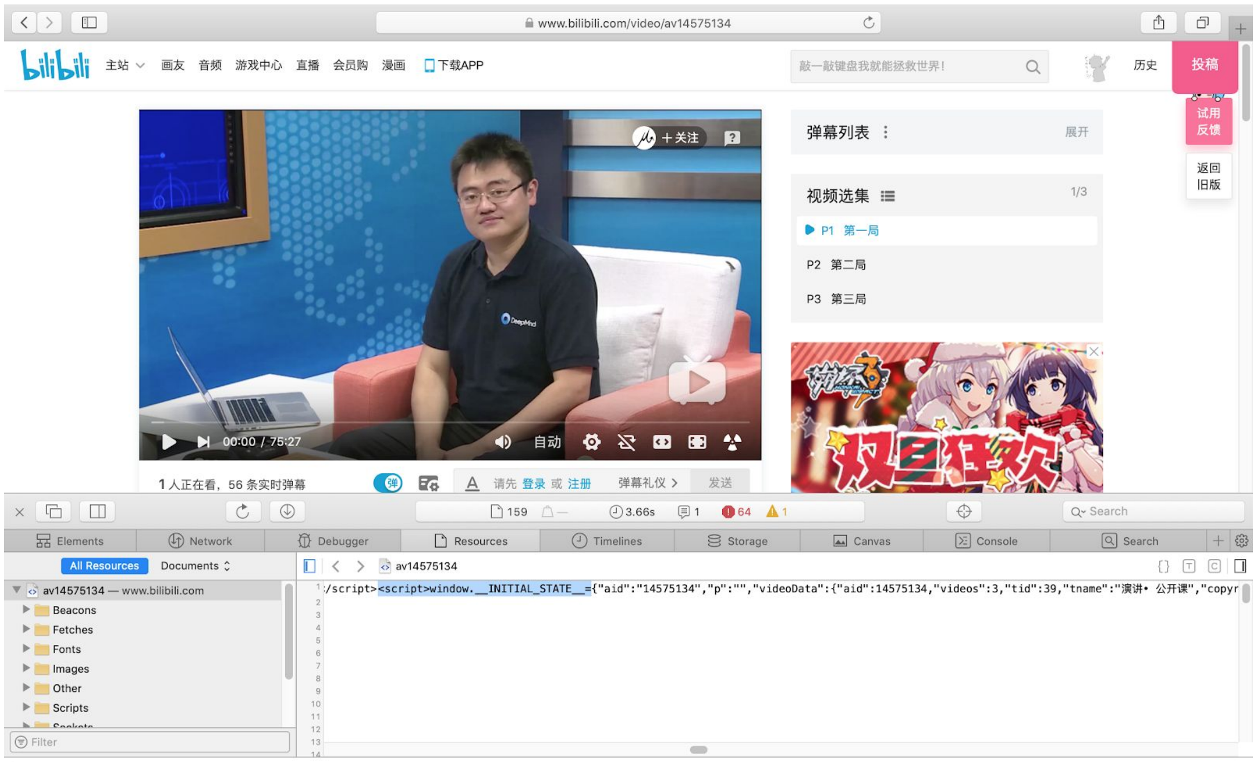
```
<a class="figure  figure-180101 "
                                                                    data-qidanadd-episode="0" data-qidanadd-channelid="17" data-qidanadd-
    data-searchpingback-elem="link" data-searchpingback-param="ptype=1-1" href="http://www.iqiyi.com/w_19ru2uv54x.html" target="_blank">
   <img alt="围棋人机大战第二季: 柯洁 VS AlphaGo 第1局（浙江卫视）" title="围棋人机大战第二季: 柯洁 VS AlphaGo 第1局（浙江卫视）" src="//m.iqiyipic.com/u



   <p class="viedo_rb">
                   <span class="icon-vInfo">01:31:24</span>
           </p>
```

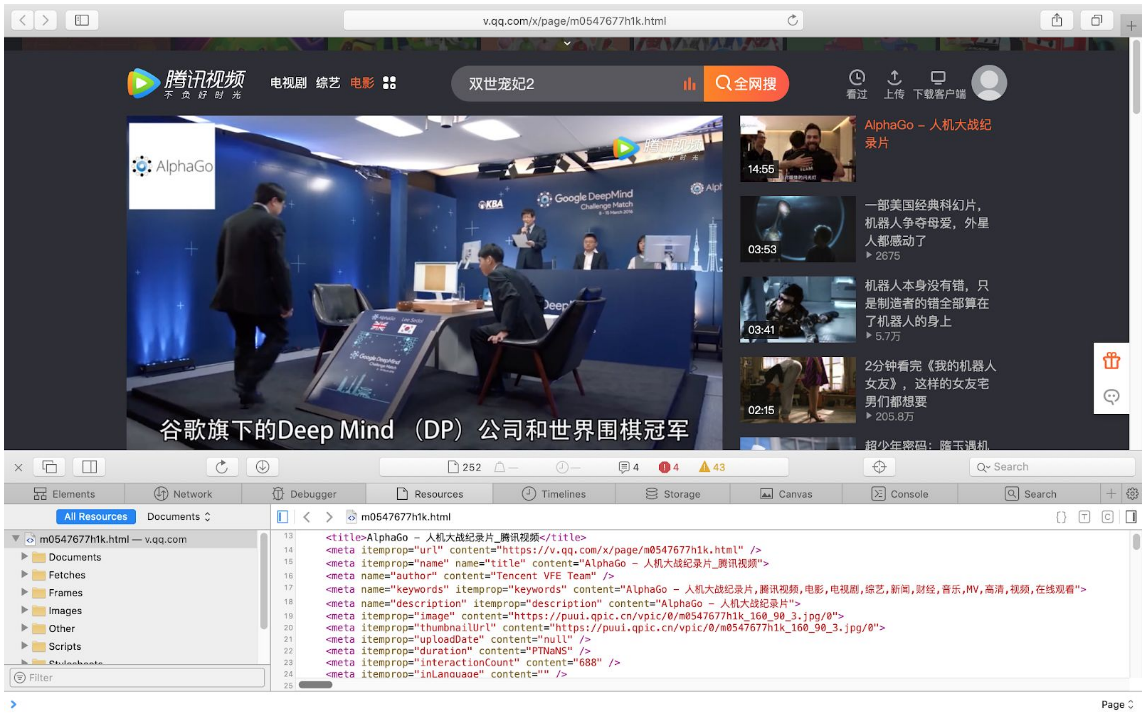
HTML Parsing – Extracting Metadata

After obtaining the URLs, we are now able to download the video. But before doing so, we would like to obtain more information about the video such as the title and description. With these metadata, we can improve queries, make further analysis and filter videos for downloading, etc. To do so, open the video URL and then open the page source. Similarly as finding the URLs in the last section, inspect the source and find the location of the metadata. We see that Bilibili videos has their information stored in a <script> tag starting with 'window.__INITIAL_STATE__'.



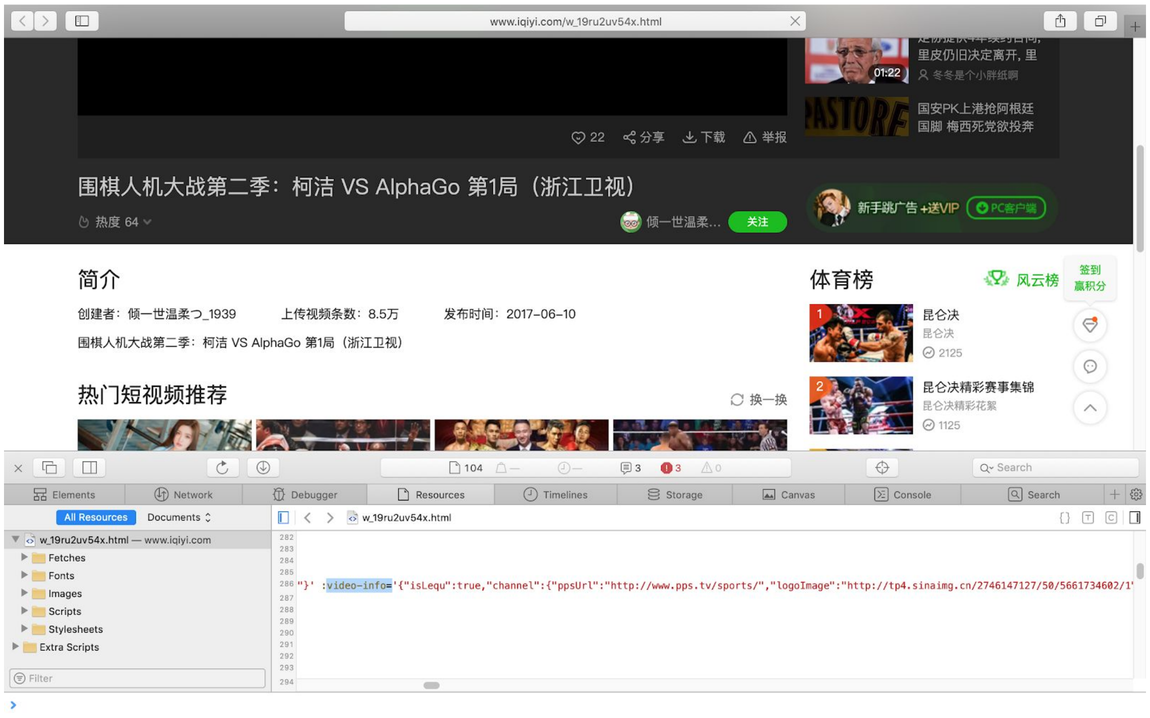
```
:/script><script>window.__INITIAL_STATE__={"aid":"14575134","p":"","videoData":{"aid":14575134,"videos":3,"tid":39,"tname":"演讲• 公开课","copyr
```

Tencent has the information in several <meta ...> tags.



```
<title>AlphaGo – 人机大战纪录片_腾讯视频</title>
<meta itemprop="url" content="https://v.qq.com/x/page/m0547677h1k.html" />
<meta itemprop="name" name="title" content="AlphaGo – 人机大战纪录片_腾讯视频">
<meta name="author" content="Tencent VFE Team" />
<meta name="keywords" itemprop="keywords" content="AlphaGo – 人机大战纪录片,腾讯视频,电影,电视剧,综艺,新闻,财经,音乐,MV,高清,视频,在线观看">
<meta name="description" itemprop="description" content="AlphaGo – 人机大战纪录片">
<meta itemprop="image" content="https://puui.qpic.cn/vpic/0/m0547677h1k_160_90_3.jpg/0">
<meta itemprop="thumbnailUrl" content="https://puui.qpic.cn/vpic/0/m0547677h1k_160_90_3.jpg/0">
<meta itemprop="uploadDate" content="null" />
<meta itemprop="duration" content="PTNaNS" />
<meta itemprop="interactionCount" content="688" />
<meta itemprop="inLanguage" content="" />
```

Iqiyi stores the information as json format in somewhere starting with 'video-info'.



```
"}' :video-info='{"isLequ":true,"channel":{"ppsUrl":"http://www.pps.tv/sports/","logoImage":"http://tp4.sinaimg.cn/2746147127/50/5661734602/1'
```

Of course, this can, and should, be done automatically. After a few manual inspections and tries Python scripts can be used to do the work efficiently.

Note that the source used for searching and the source to which the video belongs may not necessarily be the same. For example, making a query in Tencent may return resulting videos that is provided by Bilibili, Tudou, etc. Therefore, the code for obtaining URLs and the code for scraping metadata should be written separately. Also, one might notice that metadata sometimes can also be obtained from the querying results instead of going into the specific video page. However, the major downside of doing so is that the descriptions are truncated and replaced by '...', so we eventually decide to extract metadata using the more time consuming method in order to obtain the complete descriptions and other information.

Video Downloading – you-get

With no official APIs provided by the video sources, it is not easy to download the videos manually. Fortunately, there is a handy tool that can do the job, named *you-get* (https://github.com/soimort/you-get). This is a powerful tool that can download videos simply by providing the URL. It is used via the command line. For example, we can download a video by running

```
you-get https://www.bilibili.com/video/av14575134
```

in the command line. Similarly, this can also be done automatically using Python.

Before using *you-get*, we tried using another tool which is named *youkudownloader* (https://pypi.org/project/youkudownloader). However, it seems that it is outdated. Part of its code no longer works as many websites have changed the way of storing information.

Generating Queries

So far, we can easily get bunch of videos automatically, with arbitrary queries defined by us (“AlphaGo News US”, “AlphaGo News”, etc.). However, we found that the results showed up are not very accurate. In other words, there are some videos in the result which are not we target for, let say, the replay of the games Ke Jie and Lee Sedol played against AlphaGo, rather than video news. To fix this, we limit the length of the video in 5 minutes so that we can filter out the lengthy videos which are apparently not our targets. We can achieve this task by simply add an argument in pytube which support filtering on videos.

Although we can limit our videos in the length we want, we still found that the results are not as related as we expected. We thought the problem was on the keyword (query). That is to say we should put other queries which might result in more accurate videos. However, we did our best to come up with “AlphaGo News US” in human guess, which we think is the most general query to limit the results in video news which reported about AlphaGo. Hence, we decided to look into the text to find out what are the popular words which should be candidates for the query.

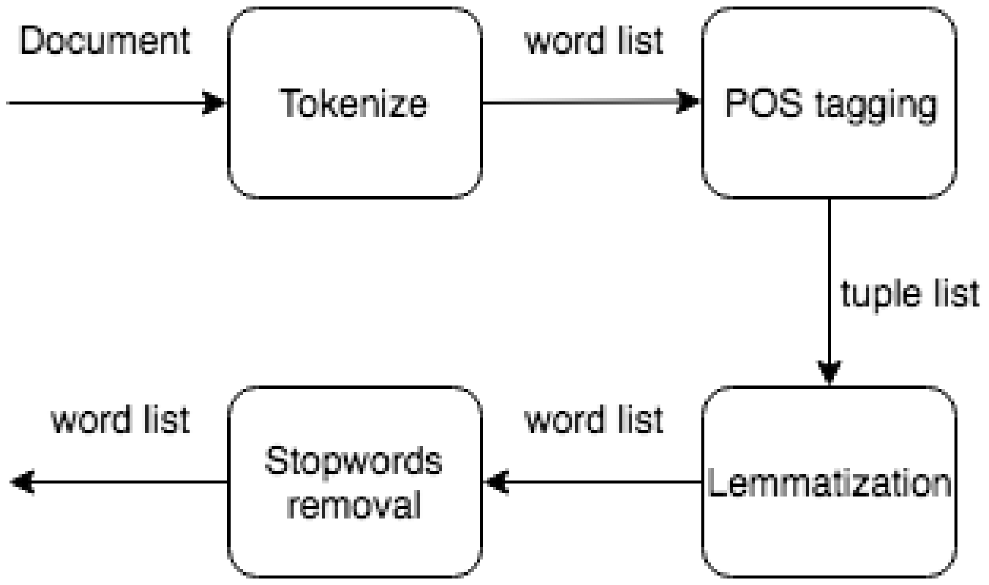
NLTK -- Document pre-processing

First, we collect the title and description of videos in the approach discussed above (Youtube API + pytube). We analyze them separately, while in same approaches. The main task of analyzing the text is to gather the word counts in each dataset (title, description). The pre-processing works of analyzing the text include some Natural Language Processing (NLP) techniques as follows:

1. Tokenization: Given a whole text file, we should chop the whole paragraph, or sentences to word level. A *token* is a sequence of characters, which should be a word in this case. This is a necessary in most cases in NLP since we usually do the analyzing in word level rather than whole sentences or characters. The input of tokenization is the title or description of the video, and the output should be a list of words.

2. Part of Speech Tagging: To give each word in the list there part of speech, for example, assign “V” to “play” in “I like to play baseball”. The reason why we do this is a preparation for the next step. The input of Part of Speech Tagging is a list of words, which should become a reasonable sentence if we join each item with spaces. The output is a list of tuple. Each tuple of the list should be (<word>, <tag>), for example, (“play”, “V”).
3. Lemmatization: There are many different forms of same words in documents for grammatical reasons. For example, “play” and “played” should be the same word in analyzing the sentences “I play baseball” and “I played baseball”. However, it should be different once the single word possesses multiple part of speech. Let’s say, “plays” in “I enjoy the perfect plays in the game”. Hence, it is clear now that the Part of Speech Tagging in previous step is necessary for the pre-processing of the documents. The input of lemmatization is a list of tuple of word and part of speech tag, and the output is a list of words since we do not need the tag anymore.
4. Stopwords Removal: A stop word is a commonly used word (such as “the”, “a”, “an”, “in”). The occurrence of theses common words will affect our result in analyzing text since these words are usually “not important” to the content of a document. The input of stopwords removal is a list of words, and the output is still a list of words while the stopwords are excluded.

All of the pre-processing steps above is implemented with NLTK (Natural Language Toolkit), a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.



Chinese Word Segmentation

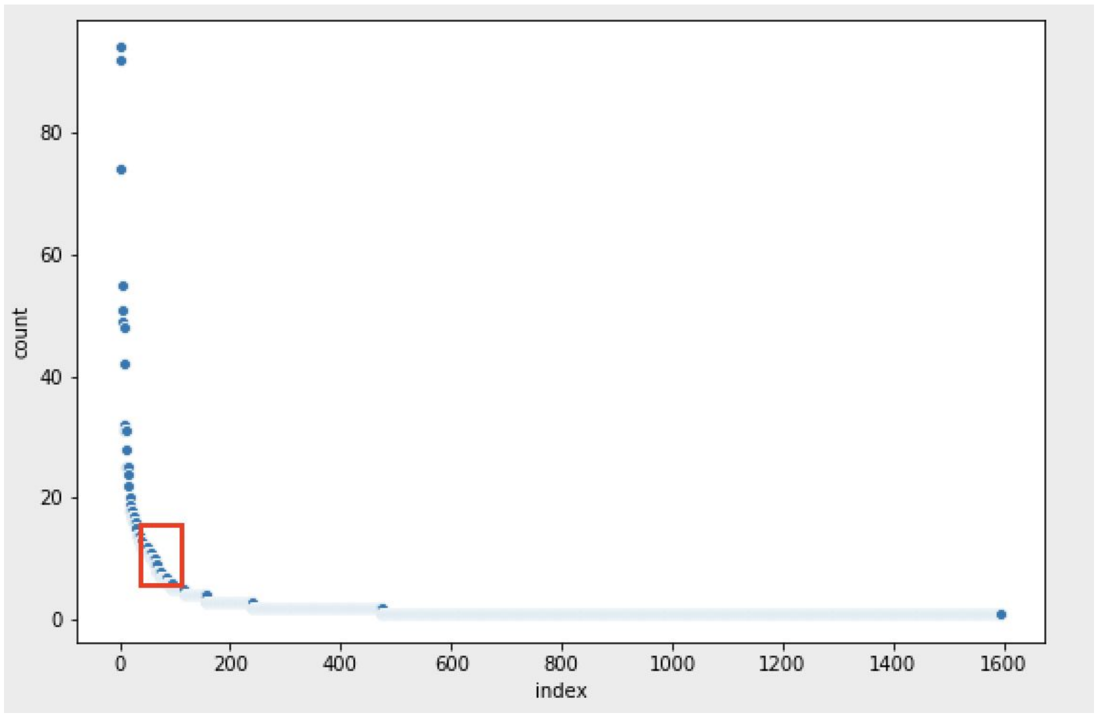
For Chinese results, there is no need to do any lemmatization, since the Chinese language does not have this type of problem. While English uses spaces as separators of words, Chinese do so according to the context. Therefore, before we can do analysis such as word count, we will need the words separated first. A nice tool for doing so is called jieba (https://github.com/fxsjy/jieba), which works fairly nice and can correctly split most of the Chinese sentences we have. The tool is written in Python and can be used via Python.

There is also a Stanford Word Segmenter (https://nlp.stanford.edu/software/segmenter.shtml) that can achieve the same purpose. But after some of our observations, we find that this tool does not perform as well as *jieba*. Many words, especially names, cannot be correctly identified by the Stanford Word Segmenter, and therefore we decide not to use it.

Look into the word counts in documents

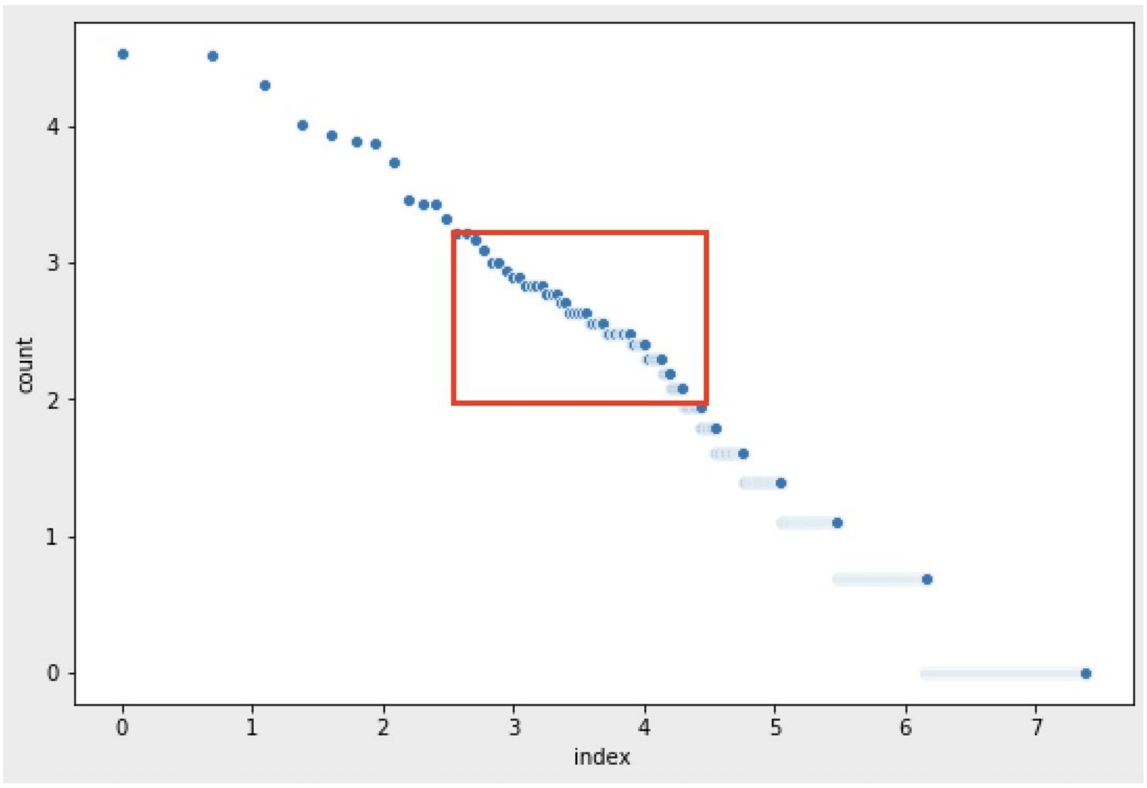
We get the processed text after implementing the steps above, then we can further enter the process of analyzing the word counts in the document. Same as before, we count the words of title and description of videos separately. In this section, we explain the process of counting the words in description of videos.

After iterating through the whole list of words and count the occurences of each of them, we found that the most popular words do not have reference value. The words are “Alphago”, “Google”, “AI”, etc., which we have already knew and tried on the query. The words we interested in is in the medium frequent range, which can be shown in the following picture.



The y-axis in the above picture is the count for each word, and x-axis is the index of the word in the list, in sorted order rather than the order occur in the document. We can see that there are few words in the head (with big number of counts), which are apparently the most popular words in the documents. And there are a bunch of words in the tail, which are also apparently the words we do not want to look into since we need the words with certain popularity in the query. Hence, we would like to extract the words from the medium popular range, which is roughly boxxed up in the picture above.

However, the curve is too deep to be readable. To solve this problem, we apply Log-log Plot to transform the deep curve into a linear line. We change the count in y-axis to $\log(count)$ and the index in x-axis to $\log(count)$. We can see the distribution of the scatters is transformed from deep curve to a linear line in the following picture.

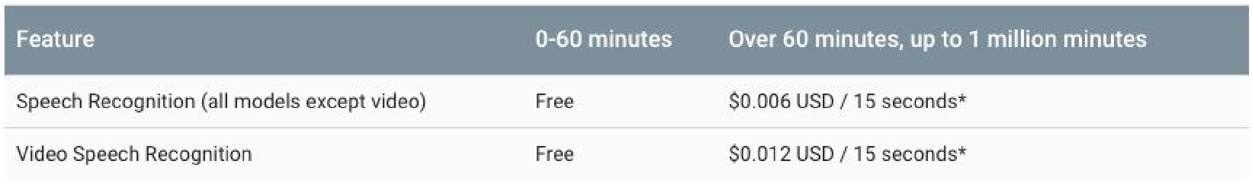


And the range of words we want to look into has become more clearer in this way.

The text part in the whole project is not limited to plain text such as titles and descriptions. However, we found that we can also work on the transcript from the anchor. Since the content in broadcast news is usually in very formal English, we can rely on common speech recognition toolkits to get the transcript.

We collected the videos with previous approach described above (Youtube API + pytube). To transcribe the text from audio, we need to convert the video file to audio file first. We use command line toolkit "ffmpeg" to do the converting in our project. We use Speech-to-Text Client Libraries by Google to transcribe our audios to text. It is a python library, while we still have to upload the audio files to Google Cloud Storage. We can then get the URL of the files and put it into the Speech-to-Text SDK. Finally, we get transcript our transcript through these steps. To avoid unnecessary cost from Google Cloud Storage, we delete the audio files right after finishing the transcript.

There is one thing we have to consider before Speech-to-Text Client Libraries by Google -- the pricing. The pricing table is as follows:

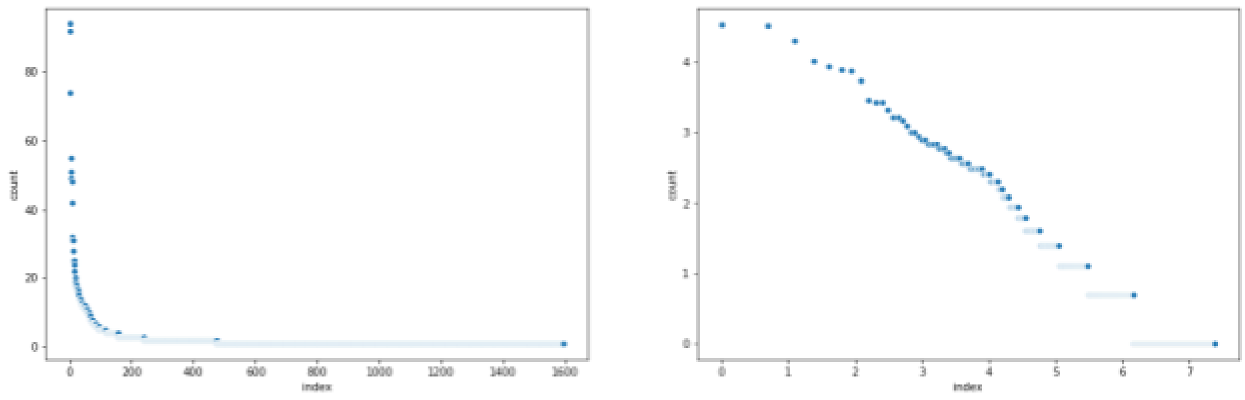


Feature	0-60 minutes	Over 60 minutes, up to 1 million minutes
Speech Recognition (all models except video)	Free	$0.006 USD / 15 seconds*
Video Speech Recognition	Free	$0.012 USD / 15 seconds*

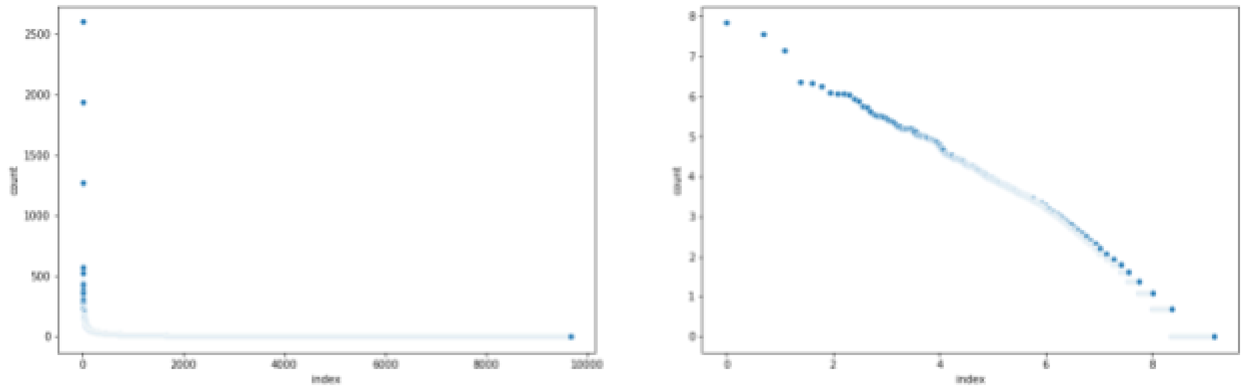
Have a note on these pricing policy could avoid unnecessary cost which might happen unknowingly.

After we get the transcript from audio files, we stored the transcripts and do the same analysis as title and descriptions. We found that in the transcript, the words tend to be more common compare to the ones in titles and descriptions.

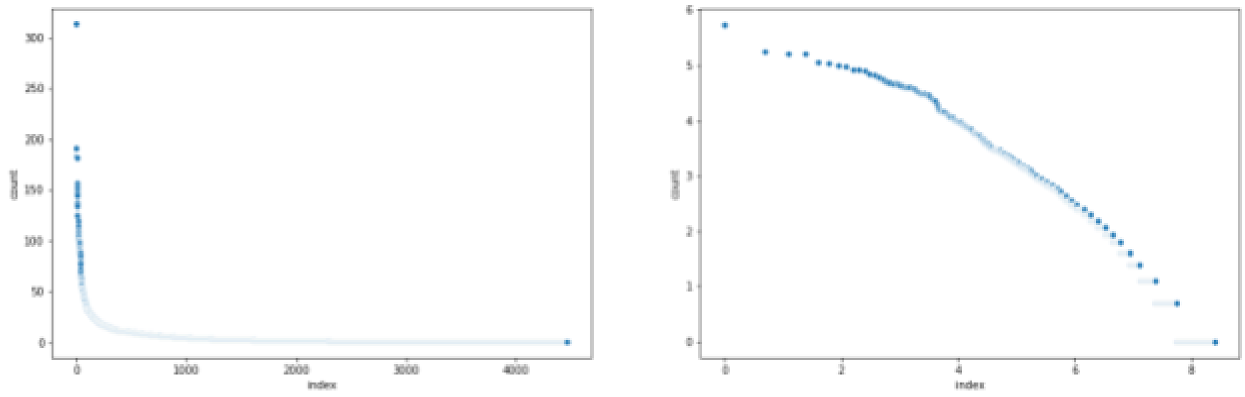
The results of the distribution of the text in normal and Log-log Plots are as follow:



Word counts on titles



Word counts on descriptions



Word counts on audio transcriptions

Here is an example of such counts (after applying segmentation).

，142		91	的 87	。72	AlphaGo 66		柯洁 47	31	在 30	日 27	
月 26	了 26	3 23	大战 23	人机 22	围棋 21	4 17	5 16	李世石 16	人工智能 15		
、14	：14	1 14	比赛 14	“ 13	谷歌 13	与 13	” 13	人类 12	进行 12		
0 11	和 10	最终 9	棋手 8	执白 8	棋 8	是 8	子 8	战胜 8	2 8		
以 8	第 7	手 7	就 7	三番 7	25 7	比 7	27 7	中国 7	浙江 6		
第二局 6	: 6	执黑 6	10 6	155 6	对决 6	— 6	至 6	！6	团队 6		
就是 5	中盘 5	中 5	自己 5	桐乡 5	第一 5	被 5	23 5	双方 5	上 5		
乌镇 5	也 5	弈 5	我 5	九段 5	- 5	下 5	胜 5	37 4	不 4		
没有 4	韩国 4	由 4	最强 4	大 4	30 4	无锡 4	13 4	世界围棋 4		对弈 4	
它 4	正式 4	对阵 4	输给 4	第二季 4	都 4	将 4	世界冠军 4		目前 4	丨 4	
前 4	但 4	狗 4	认负 4	已经 3	胜利 3	峰会 3	于 3	数据 3	秒 3		
酒店 3	并 3	？3	Deepmind 3		对战 3	著名 3	从 3	举行 3	说 3	曝光 3	
盘 3	12 3	Master 3	去 3	宣布 3	学习 3	第四场 3	世界 3	科技 3	展开 3		
DeepMind 3		6 3	首尔 3	太 3	/ 3	继续 3	排名 3	手中 3	到 3	阿尔法 3	
手机 3	结束 3	场 3	经过 3	小时 3	落败 3	发布 3	超过 3	一点 3	中国围棋 3		
分析 3	总比分 3	战成 3	时 3	再 3	了解 3	；3	分 3	战绩 3	机器人 3		
透露 2	三 2	曾 2	对 2	很 2	国家体育总局 2		曙光 2	积分 2	华美达 2	选手 2	
209 2	100 2	什么 2	本期 2	版本 2	仪式 2	聂卫平 2	意向 2	60 2	较量 2		
公布 2	抽签 2	第三局 2	年 2	反超 2	遭 2	版 2	还是 2	有过 2	高手 2		
11 2	年内 2	联想 2	局 2	挑战赛 2	库 2	成为 2	四局 2	胜率 2	开局 2		
旗下 2	最新 2	说明会 2	更好 2	一度 2	日本 2	《2	深度 2	网友 2	认输 2		
竭尽全力 2		锦标赛 2	无功而返 2		… 2	AI 2	接触 2	完美 2	们 2	冠军 2	包括 2
业余 2	运动 2	我们 2	带 2	层 2	中国围棋协会 2		由于 2	首次 2	管理中心 2		搜索 2
坚如磐石 2		号 2	阵前 2	新 2	自我 2	工具 2	其实 2	Alphago 2	初步 2	15 2	
神秘 2	惠山区 2	安 2	360 2	古力 2	抓住机会 2		届 2	近日 2	之前 2	万美元 2	
AlphaGoZero 2		等 2	达成 2	好 2	横扫 2	赢 2	论文 2	排行榜 2	意思 2	实现 2	
杨俊 2	规则 2	发展 2	不敌 2	几次 2	五场 2	战罢 2	终极 2	追 2	连败 2		
落幕 2	分钟 2	迎战 2	党委书记 2		银杏 2	你 2	得 2	需要 2	厅 2	Zero 2	
问题 2	吧 2	取得胜利 2		坦言 2	此次 2	完败 2	围棋界 2	介绍 2	看到 2	棋牌 2	
世界排名 2		》2	19 2	Google 2	后 2	智能 2	∕2	实在 2	岁 1	第一场 1	
柯洁输 1	用时 1	存储 1	时代 1	专业知识 1		出场费 1	回顾 1	秦朔 1	劣势 1	打赏 1	
树 1	中旬 1	决战 1	欲 1	这是 1	海峡两岸 1		某 1	简单 1	小 1	布局 1	
连续 1	三局 1	棋圣 1	拿 1	本局 1	现在 1	入侵 1	起飞 1	想 1	现状 1		
头筹 1	某人 1	寸 1	回事儿 1	深受 1	新主播 1	做 1	能 1	吓 1	不佳 1		
影响 1	棋子 1	硝烟 1	摩天大楼 1		夺得 1	花掉 1	一次 1	下午 1	战斗 1	早 1	
四季 1	首局 1	对局 1	③ 1	骄傲 1	顶级 1	收看 1	Pro 1	负于 1	区域 1		
主持人 1	用于 1	包 1	恶手 1	章 1	Nature 1	拔得 1	激战 1	代为 1	发表 1		
只是 1	2.0 1	战全胜 1	导读 1	iPad 1	绝艺 1	首胜 1	无 1	关乎 1	移植 1		
7 1	线 1	打破 1	秘诀 1	争霸赛 1	能下 1	④ 1	配 1	是否 1	即将 1		
击败 1	三天 1	座位 1	运营商 1	普遍认为 1		下法 1	信 1	两颗 1	博士 1	纯粹 1	
机机 1	技术 1	变化 1	天内 1	年轻人 1	下滑 1	核医学 1	② 1	让 1	第四 1		
首款 1	迪拜 1	自学成才 1		要输 1	游戏 1	电信 1	欢迎 1	alphago 1	很小 1	奖金 1	
比分 1	天 1	走向 1	落泪 1	短短 1	法国 1	出乎意料 1		体坛 1	在意 1	因为 1	

来到 1　Z 1　时态 1　究竟 1　拉开帷幕 1　传媒 1　教学 1　】 1　能力 1　安全 1
水泄不通 1　分降 1　赛后 1　43 1　跨越式 1　开战 1　10.5 1　有人 1　正常人 1　尿 1
遭遇 1　获胜 1　日与现 1　不再 1　十大 1　反应 1　计算 1　所有 1　Viv 1　自然 1
军工 1　这期 1　以为 1　① 1　病例 1　集锦 1　好玩 1　先行 1　订单 1　发布会 1
里 1　人物 1　戳 1　一年 1　原谅 1　如约 1　出 1　iPhone8 1　组成 1　识破 1
一会儿 1　之后 1　万块 1　助手 1　取得 1　使用 1　终于 1　创始人 1　科学实验 1　来 1
自学 1　万 1　等级分 1　重出江湖 1　记者 1　亿航 1　再战 1　中国棋院 1　开发 1　更 1
满足 1　尊严 1　极限 1　当 1　母亲 1　视角 1　不会 1　黑 1　泄露 1　深思 1
依旧 1　领域 1　中曾 1　万多 1　多位 1　视频 1　184 1　打败 1　柯杰 1　公司 1
再次 1　速度 1　独门 1　工程图 1　还有 1　进化版 1　在内 1　节目 1　网络 1　外形 1
其官 1　陈晖 1　希望 1　常用 1　PS 1　扳回 1　不必 1　学着 1　今天 1　先为 1
...... 1　忍不住 1　迎来 1　289 1　动态 1　认知 1　没想到 1　Alpha 1　四分之一 1　出现 1
配置 1　上市 1　不黑 1　去年 1　这么 1　大敌当前 1　谁 1　微软 1　赢得 1　柯洁能 1
或 1　吗 1　之 1　表现 1　退出 1　猛 1　全球 1　最 1　人柯洁 1　据悉 1
执子 1　现实 1　挥舞 1　Poweron 1　人们 1　紧张 1　示意 1　直接 1　一样 1　刚刚 1
三星 1　打法 1　BattleBots 1　故事 1　成员 1　晚间 1　不断 1　负 1　独自 1　第二 1
成绩 1　惜败 1　智慧 1　日至 1　着急 1　引入 1　千台 1　映射 1　柯洁用 1　通过 1
旋转 1　为 1　顶尖高手 1　开启 1　真的 1　微信 1　以往 1　第一季 1　nova 1　语音 1
大规模 1　一期 1　一部分 1　五星红旗 1　无法 1　短手 1　这样 1　成 1　前置 1　医疗界 1
当前 1　第一口 1　今年 1　研发 1　套路 1　朋友圈 1　转发 1　详情 1　工程师 1　投子 1
几度 1　中柯洁 1　力 1　深入 1　IBM 1　DeepZenGo 1　电邮 1　给出 1　随着 1　设定 1
传说 1　正史 1　他们 1　一局 1　第一期 1　一下子 1　话题 1　挤 1　多少 1　Siri 1
确认 1　像 1　第一名 1　坐不住 1　赢下 1　级 1　大赛 1　研究 1　网名 1　走 1
思路 1　变成 1　圈 1　还会 1　炫酷 1　盘负 1　上帝 1　靠着 1　提前 1　做客 1
建设 1　Note6 1　败绩 1　最近 1　一个 1　阶段 1　但是 1　, 1　会议厅 1　程序 1
很少 1　坐等 1　只 1　时间 1　40 1　看好 1　这 1　起 1　一战 1　No.1 1
战争 1　秘籍 1　责怪 1　Stone 1　重围 1　黑锅 1　知道 1　要 1　中日韩 1　摄像机 1
2017 1　成名 1　运输 1　YOGA 1　这下 1　雅森 1　大家 1　一下 1　至此 1　北京 1
电饭锅 1　凌晨 1　人狗 1　打脸 1　千年 1　以及 1　相关 1　学 1　度 1　公众 1
星际争霸 1　上演 1　胜者 1　直指 1　表情 1　谈论 1　取胜 1　小米 1　摄像头 1　媒体 1
决定 1　视 1　当场 1　令本龙 1　其中 1　进化 1　微博称 1　主播 1　科客 1　朴廷桓 1
粉丝 1　有 1　二楼 1　一新 1　~ 1　消息 1　他 1　将会 1　网上 1　⑤ 1
17 1　老将 1　然而 1　个子 1　该 1　54 1　突破 1　凭借 1　强化 1　获 1
参与 1　器官 1　摘要 1　具体 1　... 1　回复 1　打造 1　局在 1　黄世杰代 1　步棋 1
坚持下去 1　环卫工 1　Moto 1　28 1　全新 1　那版 1　广电 1　指点 1　压倒性 1　哭 1
现 1　开赛 1　【 1　名将 1　刘小光 1　模式 1　积蓄 1　完全 1　哽咽 1　黄士 1
职业 1　华为 1　天下 1　蒙特卡罗 1　记忆 1　重新 1　小学生 1　面对 1　路在何方 1　尽管 1
夜视仪 1　最酷 1　擂台 1　病患 1　常昊 1　气氛 1　杂志 1　难 1　制霸 1　真机图 1
N5s 1　排名表 1　请 1　抛弃 1　耕耘 1　啦 1　离开 1　720 1　深蓝 1　用 1
传授 1　平 1　谈 1　不妨 1　连 1　阿法 1　目标 1　留意 1　不过 1　万种 1
黑子 1　流泪 1　46 1　给 1　小编 1　意图 1　迄今为止 1　允许 1　不要 1　称 1
拥有 1　交手 1　一步 1　超越 1　地点 1　杰 1　Crazy 1　最新版 1　现代科技 1　普通 1
3D 1　诸多 1　150 1　20170528 1

Named Entities

To narrow down the range of targets what we are looking for our queries, we decide to look into named-entities. As described in WIKIPEDIA: a named-entity is a real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name. The reason why we think it is important to dive into is named-entities usually differ more apparently between different cultures.

We use a python library SpaCy for extracting the named-entities from documents. It's is an industrial level Natural Language Processing package for python which is totally free. We can just input the raw text and then output would be list of entities.

The documents we look into are descriptions. We found that there are many advertisements in the descriptions which would influence the accuracy such as "Follow us on Twitter", "Download on AppStore", etc. Hence, there are some irrelevant entities showed up in the results such as "Twitter", "AppStore", "Facebook", etc. Advertisement removal should be one of the future works which might improve the performance of our results on generating more appropriate queries.

Specific News Channels

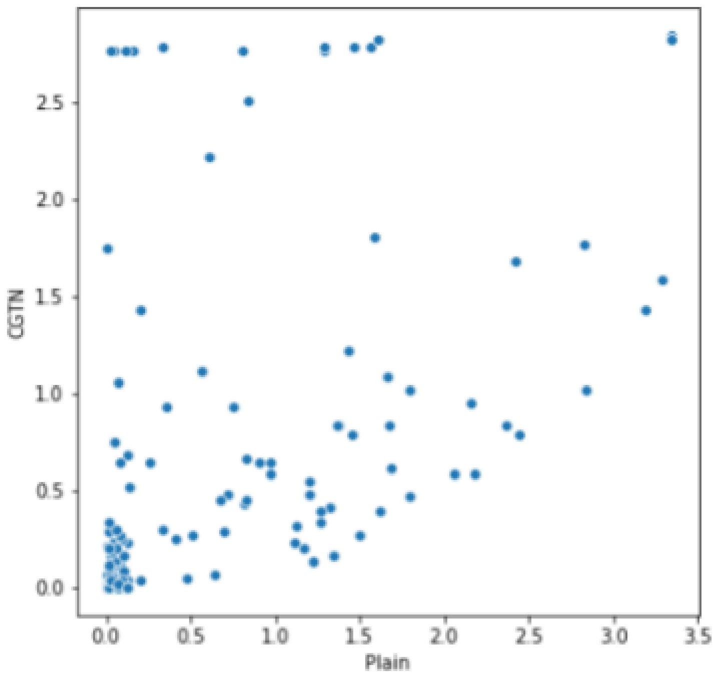
Despite the problem we met, we still found an interesting word in the result. There is an entity "CGTN" in the result that looks like a name of the news firm which might possess many resource of news about AlphaGo. We then found that CGTN stands for China Global Television Network, which is a Chinese international English-language news channel that might have videos being our targets. The reason is that we want to compare the cultural difference between China and the United States, and the feature that both videos are in English is a great help.

Hence, we turned our targets to Youtube Channels in order to narrow down the range of the results. That is to say, we queried "AlphaGo News US" in specific

Youtube Channel (more precisely, news channel) and see if the result would be more relevant to our target. Youtube API also support to search on specific Youtube channel, we only have to pass the channel ID as argument. The result turned out to be more accurate videos showed up in the first view videos.

Comparing Searching Approaches

We decided to compare the difference in results between searching in CGTN (with channel) and without specific channel. The way we compare between them is to append each named entity to "AlphaGo", that is to say, "Alphago <entity>" as the query and get the number of the results in each way. Then we normalize each data series so that we can take weight (percentage) rather than actual counts to compare with each other. We again visualize the comparison so that easily see if something strange (interesting) happens.



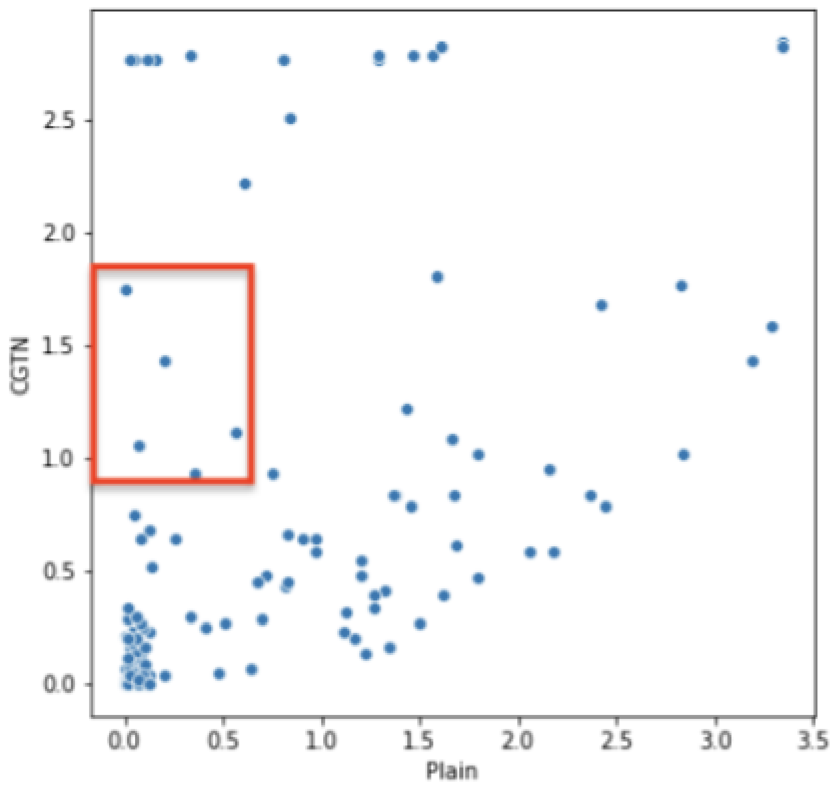
Distribution of results of search with entities in CGTN channel or general search

Each spot in the picture above represents a word (named entity). The x-axis is the percentage of the count of the word in results of searching without specific channel. The y-axis is the percentage of the count of the word in results of searching with CGTN channel.

The spots distribute in the diagonal line from lower left to upper right are the "normal cases". Which means the percentage of the word in both search approaches are similar in certain threshold range. The case we want to find is that have dramatic difference of percentage in both search approaches. In other words, the spots which appear in the upper left or lower right corners.

We can see that there are some extreme cases in the upper left corner. Those are the advertisements since specific firm might want to advertise their mobile application on Apple Store, Page on Facebook, etc., which we mentioned before. If we set both maximum and minimum thresholds of the difference of the percentage, we can find some words interesting.

The entities such as "Ke Jie" appears in the spots boxxed up above, which make sense since CGTN is a news channel based in China and Ke Jie is the Chinese Go player who played against AlphaGo.



Distribution of results of search with entities in CGTN channel or general search

We should query in more news channel on Youtube such as NBC, ABC, etc., to see if we can improve the accuracy of searching related news videos in future works.

There are some problems we found and we have not been able to solve it yet so far which are related to the degree of relevance.

1. First, we found that there are some videos which include multiple quick short news. The report of AlphaGo might be a little portion of the video, while most of the information in the clip is not helping our data analysis.
2. Another problem occurred while we were looking into the search results. We saw many "Philippines" and "ISIS" in named entities analysis, however, we are still not able to figure out what happened of here by human guess.
3. Third, the number of the videos we should take as references is hard to decide. In the previous sections, we mentioned that we download 500 videos

and metadata and do the analytical works on them. However, we found that the results become more irrelevant as approaching to the tail.

After all, we think we already have a potential solution of these problems. There are some clues that we might be able to access the “Relevance” of the results of searching with Youtube API. We can sorted the results with the order of each relevance between the video and query, while we still can not access the value right now. We consider to do it in the future and we think it is a valuable approach that we can give it a try.

Results

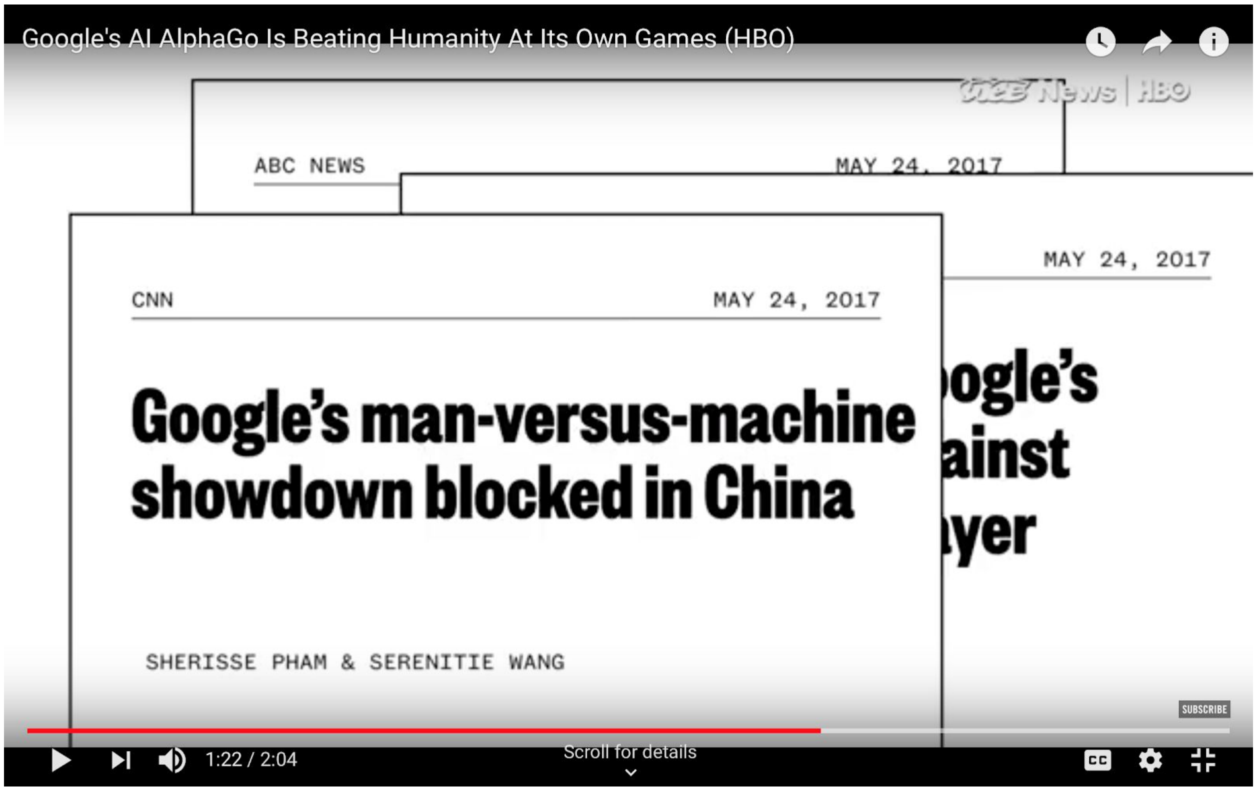
At this point, we have developed code and gathered tools so that we could download videos and obtain metadata from an arbitrary query. Currently, we observe that downloading videos from YouTube or Bilibili is very fluent. Downloading from Tencent is rather slow probably due to some internet problems. Downloading from Iqiyi is sometimes unstable and may occur errors.

Currently, our main focus is to find a way to improve our queries in order to obtain more relevant videos and discover keywords that seems to be very interesting. For example, some interesting keywords include ‘Philippines’, ‘Chess’, ‘Starcraft’, etc. These words do not usually come into mind when we think about AlphaGo. However, by doing analysis we find that these words might contain more information than we think.

Besides attempting to improve our queries in a systematic manner, we also inspect some videos manually, and here are some of the interesting results we find.

In this video from https://www.youtube.com/watch?v=8dMFJpEGNLQ, the tone towards China is very negative and treats the AlphaGo event of defeating a famous Chinese player as China’s ‘national crisis’ for Go is a traditional game in China and has a long history.

From the screenshot we see that this video states that China blocked the Google machine because of the game. However, we know that China has blocked many US sites such as Google, Youtube, Facebook, etc. already for a long time and it has nothing to do with AlphaGo.



Google's AI AlphaGo Is Beating Humanity At Its Own Games (HBO)

ABC NEWS MAY 24, 2017

MAY 24, 2017

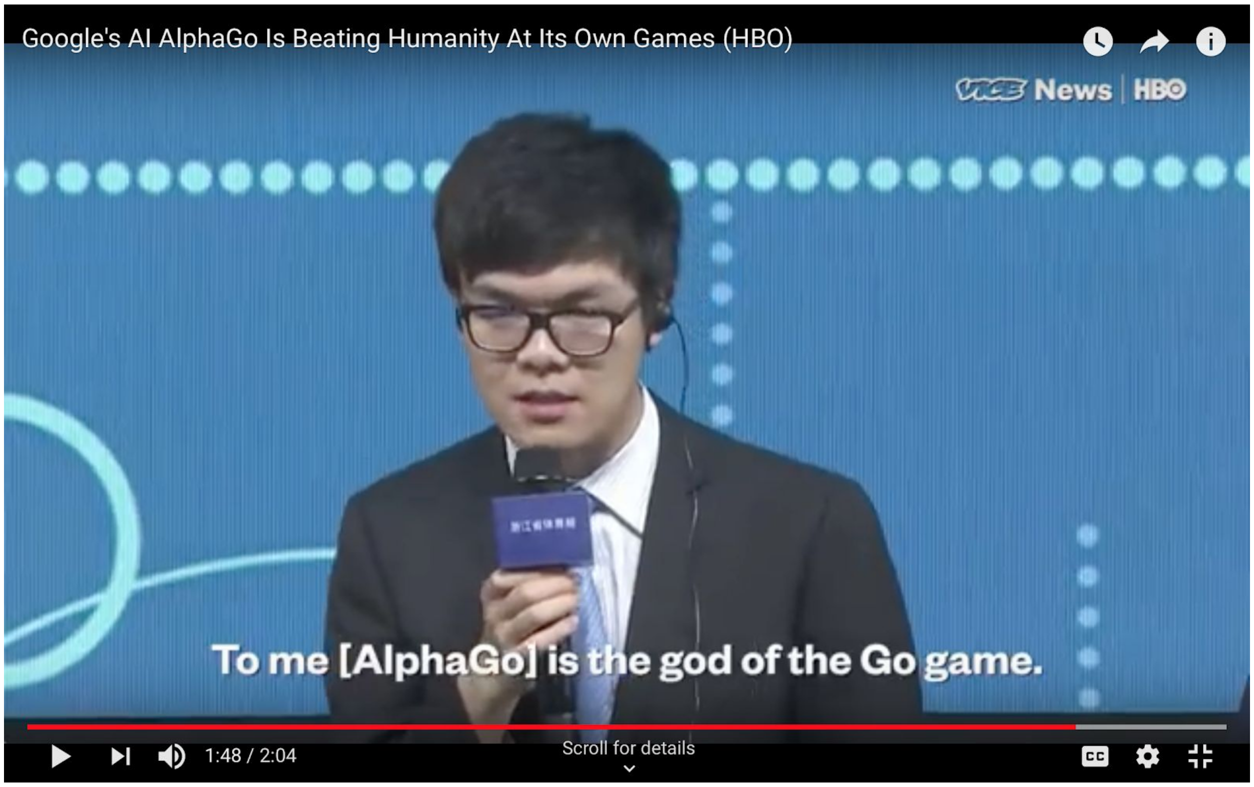
ogle's
ainst
yer

CNN MAY 24, 2017

Google's man-versus-machine showdown blocked in China

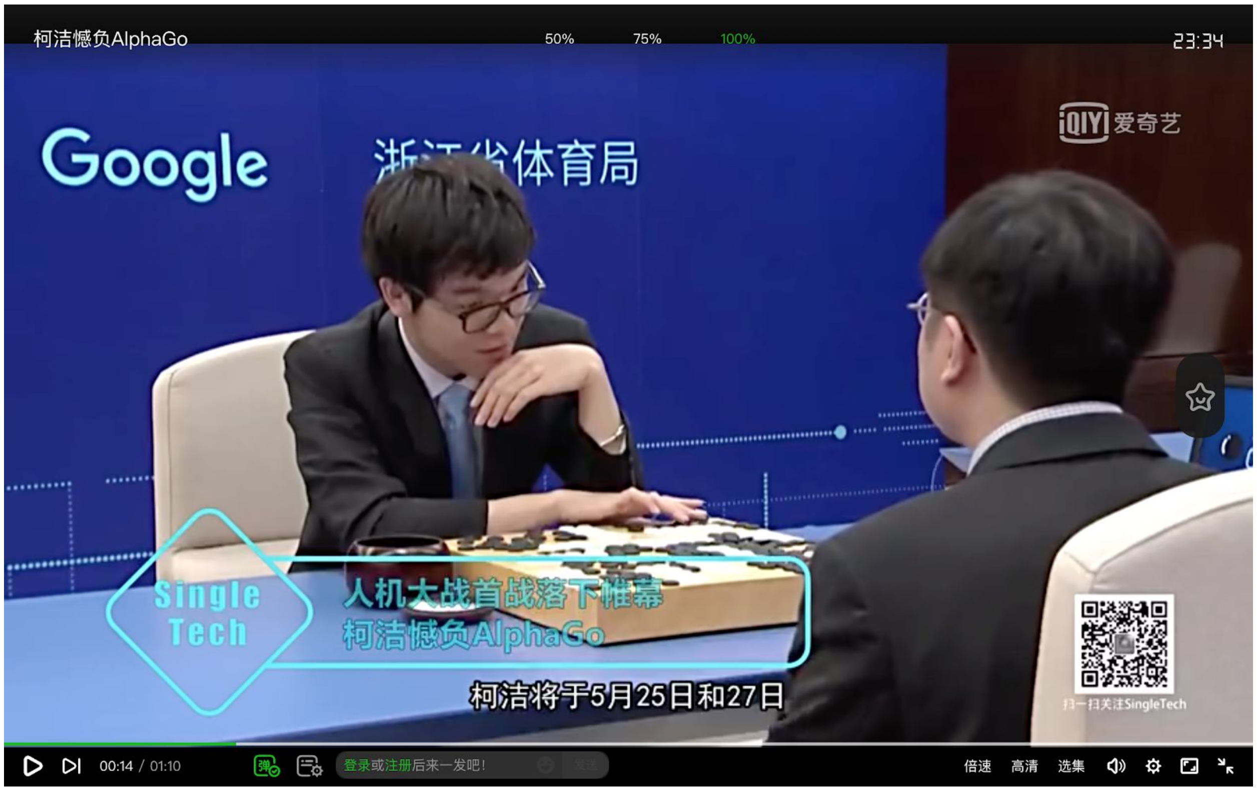
SHERISSE PHAM & SERENITIE WANG

From this screenshot we see that the video selects a small portion of the Chinese player's words and depict the feeling of depression and suffering a defeat. Although the words were really said by the player, the complete context has been removed and here these words seem to show a very different emotion from the original context.


Google's AI AlphaGo Is Beating Humanity At Its Own Games (HBO)
VICE News | HBO
To me [AlphaGo] is the god of the Go game.
1:48 / 2:04
Scroll for details

On the other hand, in a video from https://www.bilibili.com/video/av10820622?from=search&seid=15276494022864839122, we see exactly the same event, but described in a much more neutral manner. It holds no attitude about the event. Instead, it simply shows the event and let people see how the Chinese player feels and what he talks about the game.

Another video from https://www.iqiyi.com/v_19rr7es1yg.html has a very positive attitude towards the Chinese player and it describes the defeat as 'pitifully' in the title.



Therefore, we see that videos from different cultures may have significantly different opinions towards the same event, and we hope that we can improve our methods and find more such examples for further analysis.

Conclusion

We list the some ideas and future works below, including the potential solutions of the problems mentioned above:

1. Although Youtube is the most popular and we can even say it is monopoly in video platforms, we still can try to scrape videos from other sources. One of the potential benefits is we possibly can focus on news videos if we directly get the videos from news channel companies websites.
2. Query on news channels on Youtube since we can further narrow down the range of search results such as the way we query on CGTN mentioned above.
3. Irrelevant videos may still appear in the search results even if we implement many approaches in searching, while we can find other approaches to filter out those videos.
4. Removing advertisements will significantly improve the utilities of the named entities since they hold significant percentage of the counts in the documents.
5. Access the relevance score in the search result of Youtube API. This is the most important and useful way I think we can improve our accuracy on searching since we can automatically filter out the irrelevant videos with certain threshold on relevance score.