בס"ד

Advance project in computer science

YouTube Mini-Crawler Program for Video Database Construction

Personal Information

- Name: Carmi Nehoray
- ID: 065 910 267
- Address: Shtulim, HaMeyasdim 100
- Email: Nehorayc@gmail.com
- Cell-Phone: 0525287784

Supervisors names in alphabetical order:

- Dr. Avigail Mireille
- Dr. Cohen Azaria
- Dr. Lerner Anat

1. Abstract

In this project we developed a dedicated web crawler for data collection within the YouTube environment, for purpose of studying the different styles of stories reading according to the genre. The crawler is a minimized version of existing crawlers, dedicated for YouTube research, by limiting the scope of the crawler we increased the run time response and improved the relevance of the results.

As research progress so does the tools for research, electronic charts and Statistics tools for data analysis, online journals for publishing and specialized search engines for academic comparison.

Yet the Data collection process is still tedious and time consuming for some researches especially when we need human subjects or Biological data such as voice or bio-signals. Some tools were meant to bypass the problem by constructing databases of biodata such as Physionet [1] for complex bio samples or Vocalid [2] for voice samples. But the database may not be up to date or may not suit exactly to your needs.

The solution for this problem came from web crawlers, a bot whose purpose is to systematically browse the WWW and gather requested data, why gather the data "manually" and record each subject when a vast updating source of video and audio segments in the form of YouTube already exist.

By creating a simplified Crawler for data collection, a database may be rebuild or updated with ease, expediting the above process and bypassing the need for manual collection.

2. Introduction

With the development and popularity of the WWW, information has never been more accessible and in the same time more unorganized, the internet and is the largest unstructured database known to us [3].

Data mining is the extraction of relevant information from database, using parameters or predictive algorithms [3], with The aid of a Crawler software, an agent capable of analyzing and extracting information from databases, researchers may narrow down the amount of information they receive from the internet to manageable amounts, For example; organizations mine social networks to research workers relationships and identify knowledge hubs [4], and epidemiologists had mined over 79000 articles on cancer to find the connection between female parity and cancer [5].

The datamining methodologies and tools are not limited to web pages only but also to other unstructured databases such as YouTube.

YouTube is a vast source of information in audio and video based research, according to YouTube press statistics YouTube has over a billion users 300 hours of video are uploaded to YouTube every minute, Almost 5 billion videos are watched on YouTube every single day, YouTube gets over 30 million visitors per day [6].

Not only YouTube contain various clips for almost any subject you may wish to research, the site also contains valuable meta-data which was collected during the clip lifetime.

Many researches today use YouTube datamining for various subjects such as gathering opinions on events or occurrences in the world [7], or even Analyzing YouTube user personality [8], and thus researchers can utilize the information from billions of users to their need.

3. Project overview

3.1. Project description

The project's software, "YouTube – Mini crawler", allow the user to collect meta-data from a given list of YouTube videos or a search phrase, after collection the data is saved in a CSV format for further analysis

The software was written entirely in C# for ease of use and GUI capabilities for students who wish to use the system but lack the time to programmatically alter its methods.

3.2. Component overview

The system is composed of 3 phases:

- Initialization
  - GUI
  - YSP(YouTube Service provider)
- Search
  - Video request thread
  - Comment request thread
- Save
  - CSV builder
  - Comment Indexer

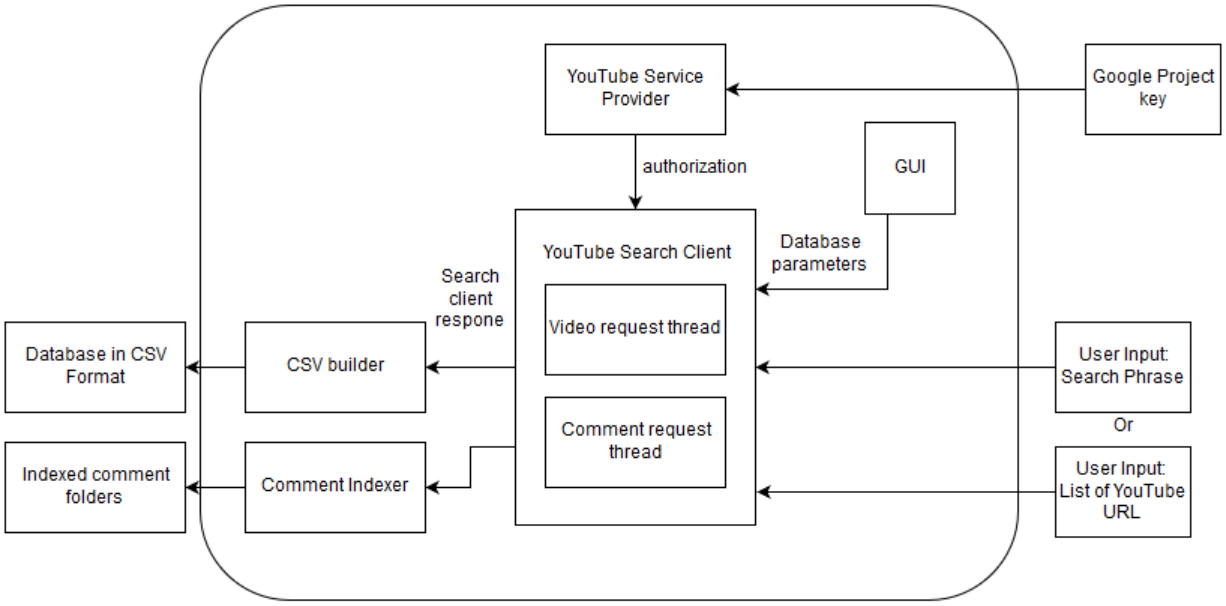

Figure 1: Crawler block diagram

3.3. Phase overview

3.3.1. Initialization

The initialization phase designation is to collect all necessary data from GUI and create the necessary YouTube server connections for the search.

The phase's functions are designed for the creation of needed Variables and services before any other phase can commence.

3.3.2. Search

Most search functions are done Via the API functions; the primary concern is the verification and preparation the search parameters and the regulation of incoming results to avoid overloading the YouTube request limit.

3.3.3. Save

The save phase collect all received data from the previous phases and store it in the requested location.

4. Tools

4.1. Google developer project

4.1.1. Review

Application programming interface or API is a set of software tools for various uses for a programmer

In March 2005 google has launched the Google developer project (GDP) or in his former name "Google code", the Google developer project was a set of APIs that allow developers to implement google functions into their software. It supports google analytics, Google earth, Google documents, Gmail, Google Maps and more. Pokémon GO for example uses the google maps for its navigation system and was actually based on Google's Annual April fools video form 2014

4.1.2. Usage

A dedicated account was created to gain full access to Google project's APIs and to provide a default key for future users without the need to open project of their own.

4.2. YouTube API V3

4.2.1. Review

The YouTube API V3 was released by google in 2014, as part of the google developer project, to allow developers access to video statistics and YouTube channels' data.

The Calls for the YouTube servers are done one of 2 ways

XML-RPC (Extensible Markup Language - Remote procedure call)

In this form the calls for the YouTube API are done by writing in XML format most suited for HTML programs and web based applications

REST (Representational state transfer)

In this form the request system of a web services (YouTube in our case) is represented in textual format allowing us to draw data and manipulate controls. This will be the form we will use in our program.

4.2.2. Usage

To use the YouTube API you must obtain a google project key, the key links directly to your YouTube account and allows you to access statistics from your own channel or from search results you made.

The API creates a search client which requires the key for identification. Once the search client is created you may use it to send calls and requests to the YouTube servers.

The data received from the API is divided to public and private data.

- Public data
    - Visible data:
    - Views
    - Upload date
    - Likes
    - Dislikes
    - Comments
    - Etc.…
- Private data
    - Views per country
    - Percentage of video watched (as users may not watch the whole video)
    - Average View Duration
    - The number of time the video was add to a playlist
    - google analytics
    - Etc.…

Public data can be accessed by anyone with a key and may also be viewd directly in the video web-page. Private data is available only to the account associated with the video and contain further analytics and information for channel managers.

Note

The API also allow users to manipulate videos and channel programmatically but as this project focus on Data collection and not manipulation this venue was not explored

5. Software overview

5.1. System description

5.1.1. Input

- A google project key
- One of the following:
    - A set of YouTube video links without their channel association (the URL must contain only the ID part of the video)
    - A search phrase and a limit on search results

5.1.2. Output

- A Log file on the operation
- If operation successful then the following will be produces:
- CSV file containing all gathered meta-data
- Indexed folders containing the comments of each entry

5.1.3. Operational requirements

- YouTube Search capability
- YouTube URL parsing
- Google server connection validation
- Data Filtering
- Save capabilities
- GUI For parameter selection
- Progress GUI
- Output for database and comments

5.1.4. Hardware requirements

- Internet connection
- Windows XP or above
- 5MB of free space

5.2. Detailed function overview

5.2.1. Initialization

5.2.1.1. initYTS

Form

```
private void initYTS(string apiKey,string searchParameters,out YouTubeService
yt,out VideosResource.ListRequest lvr,out CommentThreadsResource.ListRequest lvcr)
```

Input

apiKey – Google project key for the current user

searchParameters – the search parameters for the YouTube search clients

Output

yt – YouTube Service control

lvr – list of videos resources to be received from the search execution

lcvr - list of videos resources to be received from the search execution

Description

This function will initialize the YouTube service provider and the received parameter for the search and retrieved data. First the service will be called with the API key which will identify the user account, a default service key is provided (Email account was opened for the sake of the thesis and his key is provided).

Following the service initialization the search parameters will be set, as of now the main program set the values to "id" for specific video and "snippet" for a search phrase.

The list items for comment and video list are initialized in addition to the default tags with the "statistics" tag for additional meta-data on the video.

More tags and options can be found in the YouTube API [9] site for further development.

5.2.1.2. initCSV

Form

```
private string initCSV()
```

Input

None

Output

CSV base file according to the user specifications

Description

This function will initialize the CSV string with headlines according to the user preferences as selected in the GUI.

5.2.2. Search

5.2.2.1. Connection regulation

Although not a standalone function, the connection regulation system plays an integral role in our Software. Since we deal request the software to prepare our database from a large number of videos we risk sending too many request for the server and Block from its services.

Within the url2id function in 5.2.2.2 and line 122 in the main routine the software send the request in small portions and process each separately to avoid server block. The limit on the video request thread is 50 videos for each request (40 was taken as a precaution) and in the comment request thread only a single video request may be done at any given time.

5.2.2.2. ConIds

Form

```
private string ConIds(List<string> videoslist, int LineIndex)
```

Input

Videolist – a list of YouTube videos IDs

LineIndex – current index of the request

Output

String containing the ID of the videos for the next request

Description

The YouTube services can't process complete video URL and in order to identify a single video they need its specific YouTube ID. This function is to concentrate the list of ID's and in small groups and send it to the search list before execution.

5.2.2.3. url2id

Form

```
private string url2id(string url)
```
Input

url – a YouTube video URl address

Output

String containing the ID of the video

Description

A parser function meant to extract the ID component from a YouTube video URL

5.2.3. Save

5.2.3.1. AddCSVItem

Form

```
private void AddCSVItem(ref string CSVTable,Google.Apis.YouTube.v3.Data.Video Item)
```

Input

CSVTable – Table Containing the Videos collected database

Item – YouTube Video Item

Output

CSVTable – Table Containing the Videos collected database after update

Description

This function receive to reference to the current database we are constructing. The function extract the requested meta-data from the video item in accordance to the user preferences as specified in the GUI and store it in the table.

6. GUI interface

Screen overview

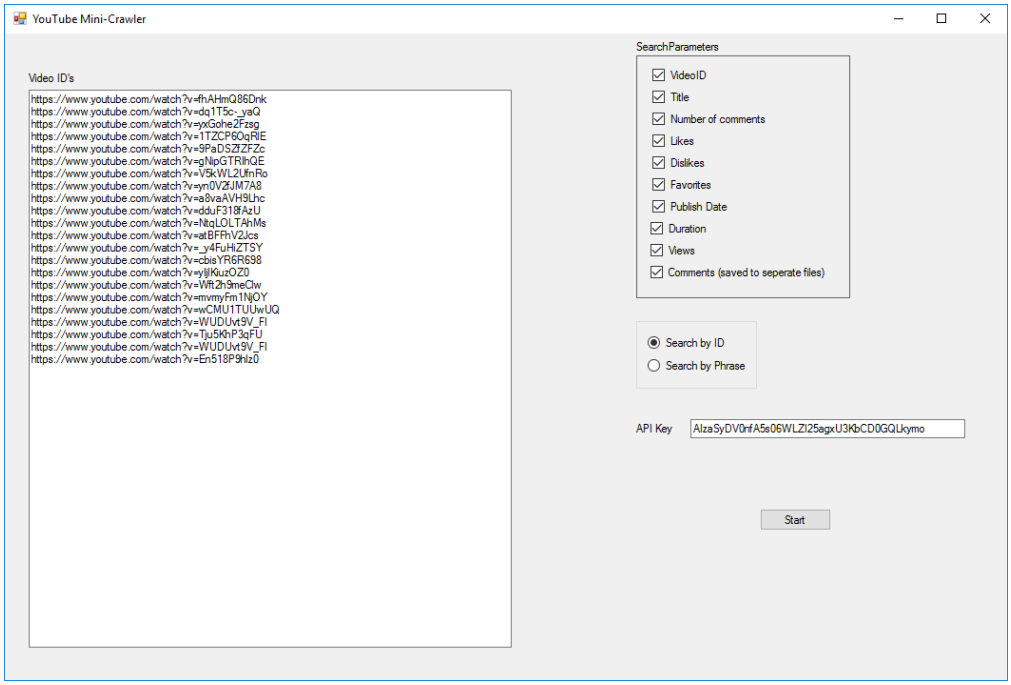

Figure 2 Main ID screen

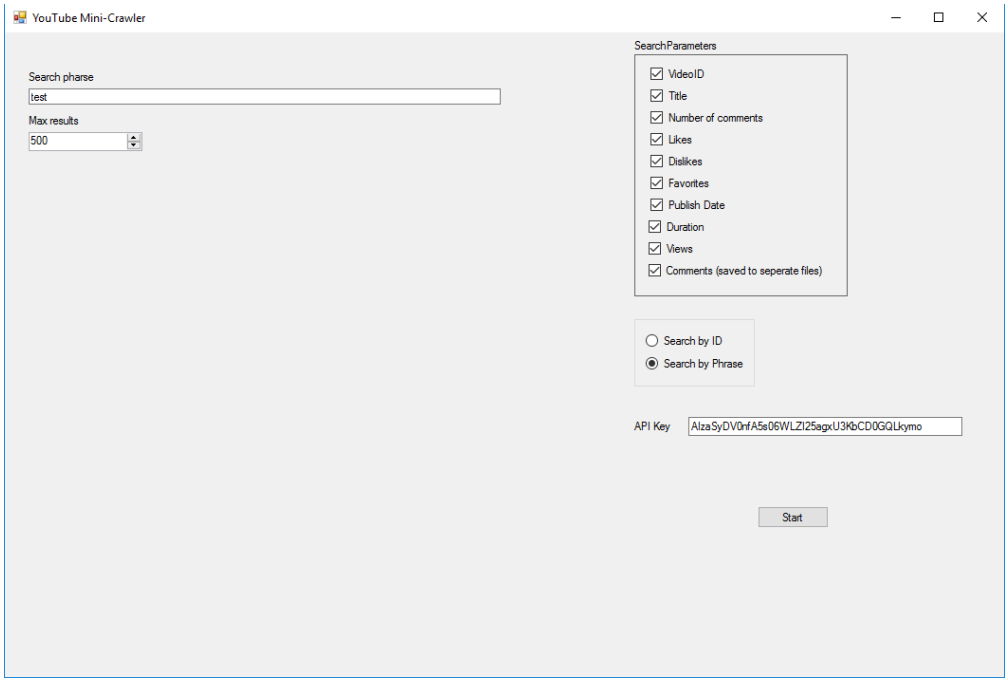

Figure 3 Main Search Phrase screen

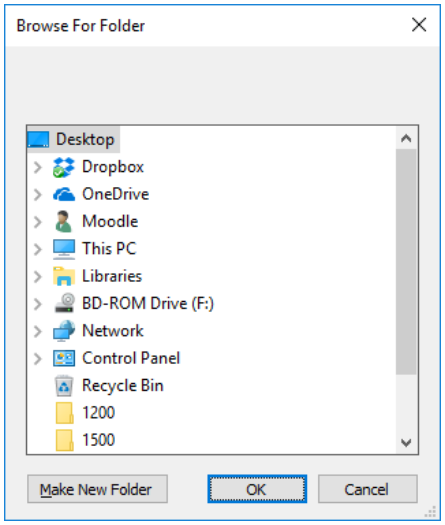

Figure 4 Folder selection

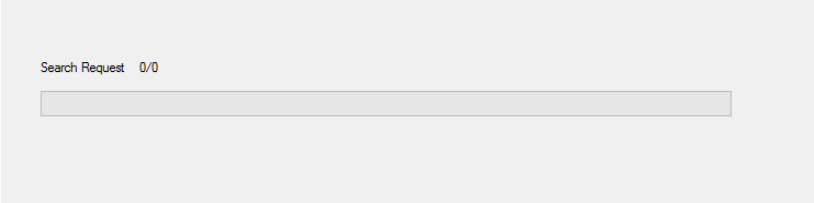

Figure 5 Progress bars

Detailed overview

API key

The API key textbox contain the Google developer key from google, the software always load a default key on startup.

Video ID Textbox

Textbox containing the user addresses for the YouTube videos to create the database.

Phrase Text box

He phrase of the YouTube query (what we are searching for)

Max results

Number of maximum results, actual number may be less because of unwanted results (playlists or channels) or lack of results

Search parameters

- Id – The YouTube id of the video
- Title – Title of the video
- Number of comments – numerical count of the comments (not the comments themselves)
- Likes – how many times the video was liked
- Dislikes - how many times the video was disliked
- Favorites – how many times people saved the video to a playlist
- Publish date – The date the video finished uploading
- Duration – total time of the video by the following format PT##H##M##S
- Views – number of times the video was viewed
- Comments – retrieve all comments as txt files indexed by the video, for example, the comments for the 3rd video will be in the comments folder in the Comments0003.txt file.

By default all parameters are extracted though the user may cancel any parameter and by this quickening the extraction process.

Folder Selection

After setting the parameters and confirming the operation the user will be asked to select a folder to save the output files.

Progress bar

As mentioned before the requests for the YouTube servers are partitioned to avoid block out, the progress bar indicate our progress in video and comment data retrieval.

7. System tests

7.1. Test case

To test the I/O 4 URLs of the little mermaid audio books were upload to the program with all search parameters

7.2. results

Data.CSV- all collected meta-data

	A	B	C	D	E	F	G	H	I
1	id	Title	Comment	Like	Dislike	Favorites	Publish Date	Duration	views
2	WUDUvt9V_FI	[FULL AudioBook] Hans Christian Andersen: The Little Mermaid	0	23	2	0	6/19/2015 7:53	PT1H1M13S	5515
3	Tju5KhP3qFU	THE LITTLE MERMAID - FULL AudioBook - by Hans Christian Anderson - Fairy Tale	17	121	48	0	12/19/2012 23:06	PT58M50S	54294
4	WUDUvt9V_FI	[FULL AudioBook] Hans Christian Andersen: The Little Mermaid	0	23	2	0	6/19/2015 7:53	PT1H1M13S	5515
5	En518P9hlz0	The Little Mermaid by Hans Christian Andersen | Children's Audiobook | Full Unabridged AudioBook	0	1	0	0	6/5/2015 10:38	PT1H11M1S	120

Figure 6 Data.CSV

4 Comments####.txt files for each entry

```
I like this book
----------------------------------------
I really  wanted to  watch this,
----------------------------------------

----------------------------------------

----------------------------------------
Ugly
----------------------------------------
Wait the prince fell in love with a fake....why did she kiss the fake on the head at the end....
----------------------------------------
The thing that first struck me about the image is that she doesn&#39;t have any sand stuck to her elbows.
She also doesn&#39;t appear to be wet.  I have no problem with the reading and can&#39;t figure out what
might possible be wrong with it.  Is it that the reader has a slight accent?  Doesn&#39;t everybody?
----------------------------------------
So if her grandmother didn&#39;t tell her romantic lies she wouldn&#39;t have gone looking for them? Poor
innocent soul.
----------------------------------------
This is very badly read.  Shocking!
----------------------------------------
Why can&#39;t people just appreciate good original literature?
----------------------------------------
WF )
----------------------------------------
Ewwwww! Really!?
----------------------------------------
```

Figure 7 Comments0001.txt

In some files the comments were disabled and the files only contained "Comments retrieval disabled"

8. Research Possibilities

The Software as of now allows to Generate and organize data based on a series of YouTube video addresses or a search word, but access is only given to the public data since we are not the owners of those videos.

A possible research methodology would be the creation of a dedicated YouTube channel with pre-made videos, and allow access for users and subjects to for a limited time, The data will be gathered automatically by the YouTube engine and collected by the researcher with the software. A channel may even be send to subjects with instructions like questionnaires are sent today.

Since the dedicated account allows full control of said channel, access to additional information from the private data is available.

Such Level of detail in the Meta-data can open new research possibilities for culture preferences via the "Views per country" parameter or a test of user patience via the "Average View Duration". That way instead of questioners that measure individual behavior and later compile their data to cultural behavior, YouTube provide us with a powerful statistical tool to analyze the user's behavior and actions during the view of the video clip directly and not by answering questions after the case scenario has ended.

9. Summery and future plans

This project is the perfect example for "Work of necessity", its development has risen from the need to collect data from dozens of YouTube audio books, it was concluded that a few days or weeks of development of such tool are favorable to data collection which cannot be repeated without spending the same amount of time and resources. .

Various tools and solution were attempted until the YouTube API was suggested.

After the initial phases of planning and defining the project it's was first constructed only as a code segment that can only be altered programmatically but in light of its added value to the data collection process and the time it can save for new researchers, a GUI and fine-tuning mechanism for search parameters were added.

Although the project is big in scope or require specially developed algorithms, it is still powerful not because of its complexity or its invitation but because of its simplicity and ease of use as an important research tool.

Future releases may delve further into the YouTube API capabilities and extract more relevant data or possibly upgrade the software to adapt to new versions of the YouTube. API

10. Legal notice

The main reason Google has updates its API from V2 to V3 was because V2 could extract video or audio segment directly from YouTube.

Such action is a breach of conduct regarding YouTube Terms of service [10] as many videos on YouTube are protected under copyright to a different degree.

To analyze audio for research purposes one must use, as we did on the thesis, the "On the fly" approach and analyze the audio as it plays from the speakers and under no circumstances save the original soundtrack.

11. Bibliography

[1] A. L. Goldberg, R. G. Mark and G. B. Mo, "PhysioNet," [Online]. Available: https://www.physionet.org/. [Accessed 17 1 2017].

[2] VocaliD , "VocaliD," [Online]. Available: https://www.vocalid.co/. [Accessed 17 1 2017].

[3] P. Ruchika and B. Pooja , "A Survey on Semantic Focused Web Crawler for Information Discovery Using Data Mining Technique," *IJIRST - International Journal for Innovative Research in Science & Technology,* vol. 4, no. 7, 2014.

[4] M. Fire and R. Puzis, "Organization Mining Using Online Social Networks," *Networks and Spatial Economics,* vol. 16, no. 2, p. 545–578, 2016.

[5] G. Tourassi, . H.-J. Yoon, S. Xu and X. Han, "The utility of web mining for epidemiological research: studying the association between parity and cancer risk," *Journal of the American Medical Informatics Association,* vol. 23, no. 3, pp. 588-595, 2016.

[6] Google, "Statistics - YouTube," [Online]. Available: https://www.youtube.com/yt/press/statistics.html. [Accessed 15 1 2017].

[7] A. Severyn, A. Moschitti, O. Uryupina, B. Plank and K. Filippova, "Multi-lingual opinion mining on youtube," *Information Processing & Management,* vol. 52, no. 1, pp. 46-60, 2016.

[8] C. Khosla, "YOUTUBE DATA ANALYSIS USING HADOOP PhD diss," California State University, Sacramento, 2016.

[9] Google, "Search - YouTube Data API," 2 November 2016. [Online]. Available: https://developers.google.com/youtube/v3/docs/search.

[10] Google, "YouTube API Services Terms of Service," [Online]. Available: https://developers.google.com/youtube/terms/api-services-terms-of-service. [Accessed 15 1 2016].

12. Figure List