UFSM

InterAção

OPEN ACCESS

IA., Santa Maria, v. 16, n. 4, e92749, p. 1-13, Oct./Dec. 2025 • https://dx.doi.org/10.5902/2357797592749
Submitted: 6th/07/2025 • Approved: 21st/08/2025 • Published: 30th/09/2025

Artigos Dossiê

Isabela Rocha [I]

Ergon Cugler de Moraes Silva [II]

YouTubeScrap: a comprehensive tool for scraping YouTube data and transcript

YouTubeScrap: uma ferramenta abrangente para raspagem de dados e transcrições do YouTube

Abstract:

YouTubeScrap is an open-source tool that streamlines the collection, analysis, and organization of YouTube video data and transcripts, tailored to researchers, analysts, and content creators. Designed for accessibility and efficiency, this tool enables users to perform targeted searches, extract detailed metadata, and retrieve multilingual transcripts without requiring API keys-addressing growing restrictions on data accessibility. Operating seamlessly within Google Colab, YouTubeScrap leverages a cloud-based infrastructure to eliminate installation barriers, offering a ready-to-use environment for users with varying technical expertise. The tool integrates Python libraries such as yt_dlp, YouTubeTranscriptAPI, and scrapetube to automate video searches, filter results by criteria like keywords and date ranges, and store outputs in Google Sheets for easy collaboration and compliance with international data privacy standards. This API-free solution democratizes access to digital content, enabling large-scale data collection and analysis across academic research, media studies, and communication fields. By simplifying complex data-handling processes, YouTubeScrap empowers users to navigate vast digital landscapes ethically and efficiently, promoting equitable access to critical online information in an era of increasing platform restrictions. This tool serves as a scalable, user-friendly resource for engaging and advancing data-driven research.

Keywords: Data scraping; Computing; Computational social sciences

Resumo:

YouTubeScrap é uma ferramenta de código aberto que simplifica a coleta, análise e organização de dados de vídeos e transcrições do YouTube, projetada para pesquisadores, analistas e criadores de conteúdo. Desenvolvida para ser acessível e eficiente, esta ferramenta permite que os usuários realizem buscas direcionadas, extraiam metadados detalhados e recuperem transcrições multilíngues sem a necessidade de chaves de API - resolvendo as crescentes restrições ao acesso a dados. Operando perfeitamente no Google Colab, o YouTubeScrap utiliza uma infraestrutura baseada em nuvem para eliminar barreiras de instalação, oferecendo um ambiente pronto para uso, adequado para usuários com diferentes níveis de conhecimento técnico. A ferramenta integra bibliotecas Python, como yt_dlp, YouTubeTranscriptAPI e scrapetube, para automatizar buscas de vídeos, filtrar resultados por critérios como palavras-chave e intervalos de datas, e armazenar os resultados no Google Sheets, facilitando a colaboração e garantindo conformidade com padrões internacionais de privacidade de dados. Esta solução sem uso de APIs democratiza o acesso ao conteúdo digital, permitindo a coleta e análise de dados em larga escala para pesquisas acadêmicas, estudos de mídia e comunicação. Ao simplificar processos complexos de manipulação de dados, o YouTubeScrap capacita os usuários a navegar por vastos ecossistemas digitais de maneira ética e eficiente, promovendo um acesso mais equitativo a informações críticas online em uma era de restrições crescentes nas plataformas. Essa ferramenta se destaca como um recurso escalável e fácil de usar, ideal para fomentar e avançar pesquisas baseadas em dados.

Palavras-chave: Data scraping; Computação; Ciências sociais computacionais

[I] PhD Candidate, University of Brasília ROR, Brasília, Federal District, Brazil.
isabelarocha.contato@gmail.com, https://orcid.org/0000-0001-8488-5528

[II] Master in Public Administration and Government from Fundação Getulio Vargas; Member in the Council for Sustainable Economic and Social Development, Brasília, Federal District, Brazil.
contato@ergoncugler.com, https://orcid.org/0000-0002-5753-1705

INTRODUCTION

Video-sharing platforms like YouTube have become an integral part of everyday life, serving as a hub for information, entertainment, and communication. Its widespread use offers researchers, analysts, and content creators unique opportunities to explore public discourse, monitor trends, and examine patterns of digital engagement. As of July 2024, the countries with the largest YouTube user bases include India, leading with approximately 476 million users, followed by the United States with around 239 million users. Brazil ranks third with about 144 million users, while Indonesia and Mexico have approximately 139 million and 83 million users, respectively (Statista, 2024). As of the third quarter of 2023, the platform was used by 91.2% of the country's online audience, ranking just below WhatsApp (93.4%) and Instagram (91.2%) (Statista, 2023). However, extracting and organizing meaningful data from such platforms can be time-consuming and technically challenging, especially when dealing with large datasets or multilingual content.

YouTubeScrap was developed to address this gap. This open-source, API key free tool simplifies the process of collecting, analyzing, and storing YouTube video data and transcripts, making it accessible to a wide audience, offering a scalable and efficient solution for gathering detailed video metrics, retrieving multilingual transcripts, and integrating the data directly into Google Sheets for further analysis. With its user-friendly design and versatile features, **YouTubeScrap** has applications across academic research, media analysis, communication studies, and digital marketing.

The tool automates various processes to maximize efficiency and accuracy. It enables users to perform targeted YouTube searches with customizable queries, filter results by date ranges, and save video links for analysis. **YouTubeScrap** extracts comprehensive video metadata, including titles, views, likes, upload dates, channel information, and tags. Additionally, it retrieves multilingual transcripts for videos that provide subtitles, facilitating content analysis across different languages. All collected data is seamlessly organized into a Google Sheet, ensuring accessibility, shareability, and compatibility with additional processing tools. By integrating Python libraries like yt_dlp, YouTubeTranscriptAPI, and scrapetube, the tool ensures flexibility. Built specifically for use in the Google Colab environment with Google Sheets, **YouTubeScrap** ensures ease of deployment and accessibility for users with varying levels of technical expertise as Colab's cloud-based platform eliminates the need for local installations and offers a ready-to-use Python environment. The step-by-step execution format allows users to interact with the tool intuitively, running functions in sequence without needing deep programming knowledge.

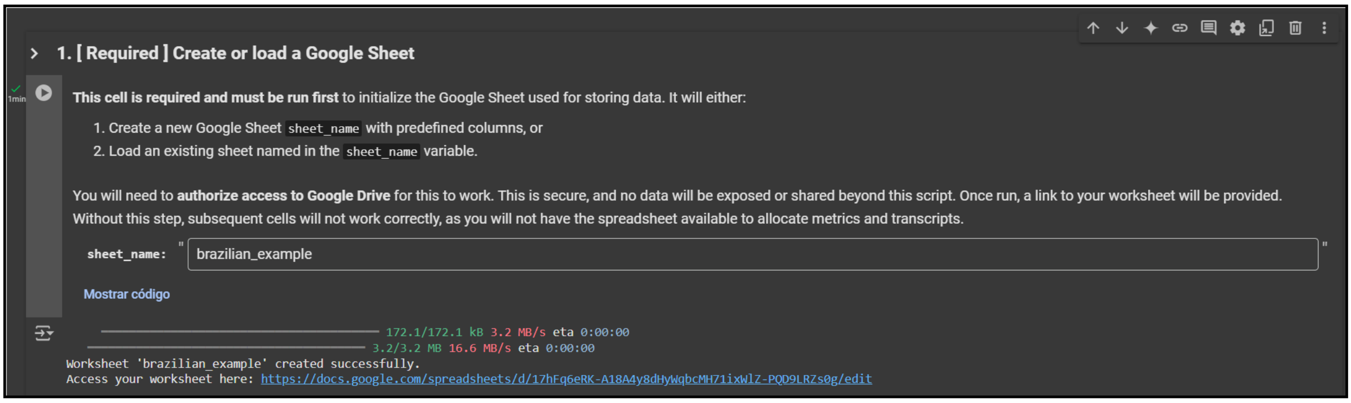
This whitepaper outlines the development and capabilities of **YouTubeScrap**, providing guidance on how users can leverage this tool to efficiently navigate and analyze YouTube content. Beyond this introductory section, the whitepaper will count with further three: the **Step-by-step usage** section provides step-by-step instructions on how to use the tool for searching videos, extracting metrics, and retrieving transcripts; the **Code** section delves into the technical aspects, explaining the functionality of key modules and offering insights for customization; and finally, the fourth and final **Conclusions** sections discuss the broader implications of the tool and potential areas for further development or application.

STEP-BY-STEP USAGE

I. [Required] Create or load a Google Sheet

Open the tool in Google Colab and evaluate the first cell where you're invited to Create a Worksheet. After authenticating your Google account to connect with Google Sheets, where the data will be stored, proceed to name your worksheet. If the specified sheet doesn't exist, the tool will create one automatically with pre-defined column headers. These headers ensure that all extracted data, such as video links, metadata, and transcripts, is structured for easy organization and analysis. Once this step is complete, a link to your worksheet will be provided for quick access and verification.

Figure 1 - First cell to be evaluated



1. [Required] Create or load a Google Sheet

This cell is required and must be run first to initialize the Google Sheet used for storing data. It will either:

1. Create a new Google Sheet `sheet_name` with predefined columns, or
2. Load an existing sheet named in the `sheet_name` variable.

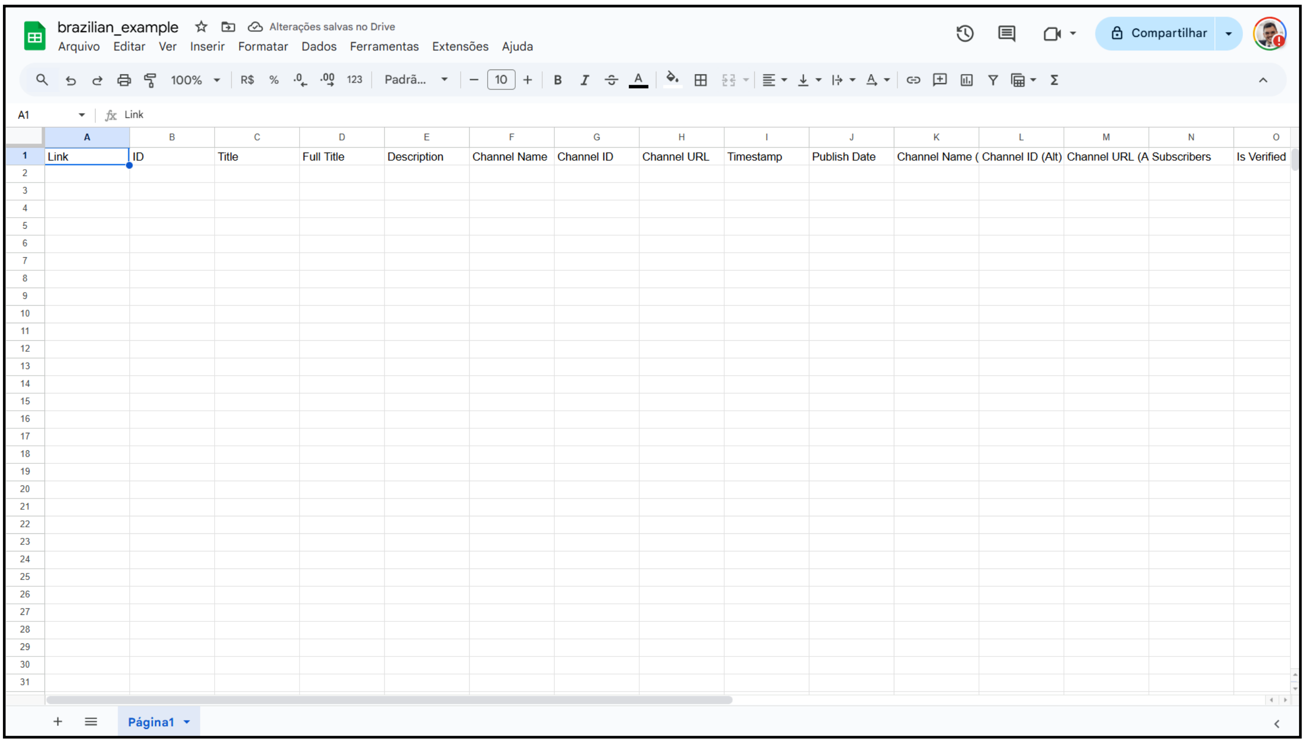
You will need to **authorize access to Google Drive** for this to work. This is secure, and no data will be exposed or shared beyond this script. Once run, a link to your worksheet will be provided. Without this step, subsequent cells will not work correctly, as you will not have the spreadsheet available to allocate metrics and transcripts.

sheet_name: " brazilian_example "

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 172.1/172.1 kB 3.2 MB/s eta 0:00:00
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 16.6 MB/s eta 0:00:00
Worksheet 'brazilian_example' created successfully.
Access your worksheet here: https://docs.google.com/spreadsheets/d/17hFq6eRK-A18A4y8dHyWqbcMH71ixWlZ-PQD9LRZs0g/edit
```

Source: Authors (2024)

Figure 2 - Result after running the first cell of the code



brazilian_example
Alterações salvas no Drive
Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda
Compartilhar
100%
R$ % .0 .00 123 Padrã... 10
A1
Link
Página1

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Link	ID	Title	Full Title	Description	Channel Name	Channel ID	Channel URL	Timestamp	Publish Date	Channel Name (	Channel ID (Alt)	Channel URL (A	Subscribers	Is Verified

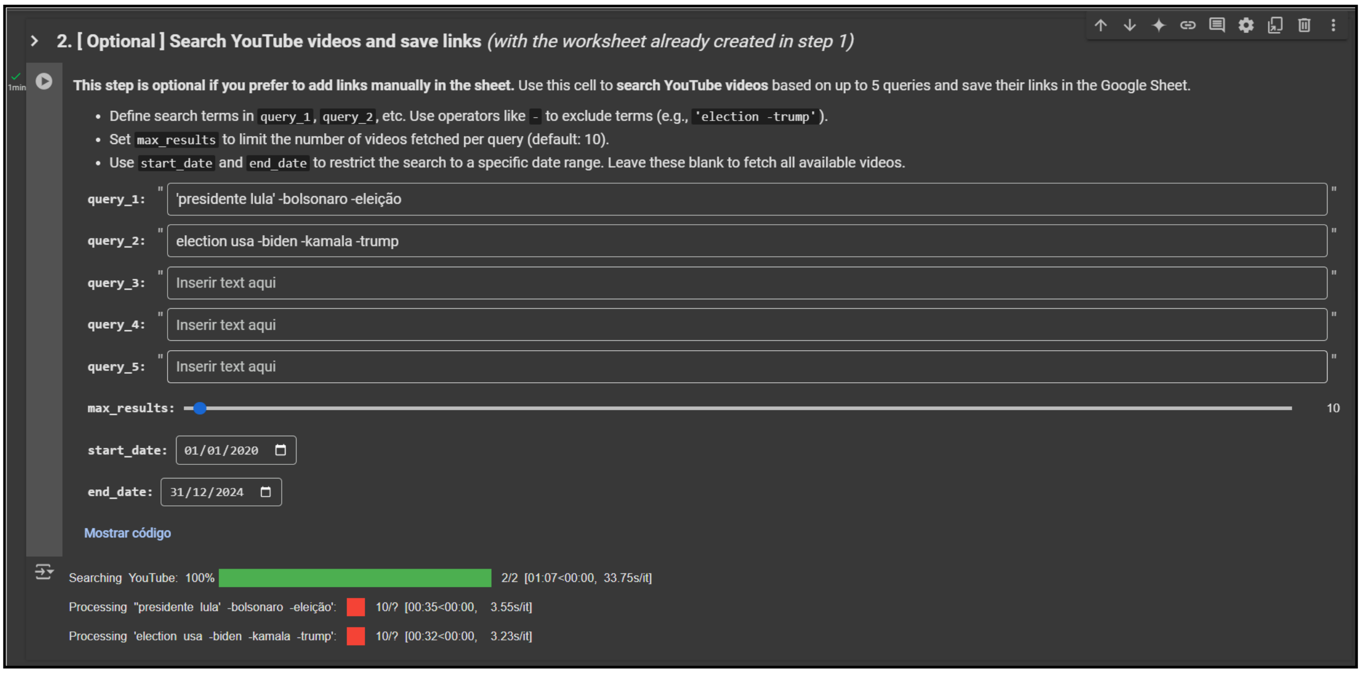
Source: Authors (2024)

II. [Optional] Search YouTube videos and save links

Define your search queries by specifying the the search queries for the YouTube videos you want to retrieve. The tool provides five query input fields (query_1 to query_5), where you can enter keywords or phrases to guide your search. Each query can include filtering options using positive or negative keywords (e.g., "presidente lula" -bolsonaro -eleição). You can also Adjust the max_results slider to set the number of videos to retrieve for each query and specify a start_date and end_date to filter videos based on their upload dates. Once you've entered your queries and filters, run this section. The tool will begin searching YouTube for videos matching the defined criteria, displaying progress for each query. The retrieved video links will be saved directly to your Google Sheet.

After gathering the data, your worksheet will contain every discovered link. On the next step, we shall add relevant Metadata to the discovered videos. **Attention: Instead of running this cell, you can also manually add links in your generated worksheet, as many as you want, to extract metrics and transcripts in the following cells.**

Figure 3 - Second cell with search queries



2. [Optional] Search YouTube videos and save links *(with the worksheet already created in step 1)*

This step is optional if you prefer to add links manually in the sheet. Use this cell to **search YouTube videos** based on up to 5 queries and save their links in the Google Sheet.

- Define search terms in `query_1`, `query_2`, etc. Use operators like `-` to exclude terms (e.g., `'election -trump'`).
- Set `max_results` to limit the number of videos fetched per query (default: 10).
- Use `start_date` and `end_date` to restrict the search to a specific date range. Leave these blank to fetch all available videos.

query_1: 'presidente lula' -bolsonaro -eleição

query_2: election usa -biden -kamala -trump

query_3: Inserir text aqui

query_4: Inserir text aqui

query_5: Inserir text aqui

max_results: 10

start_date: 01/01/2020

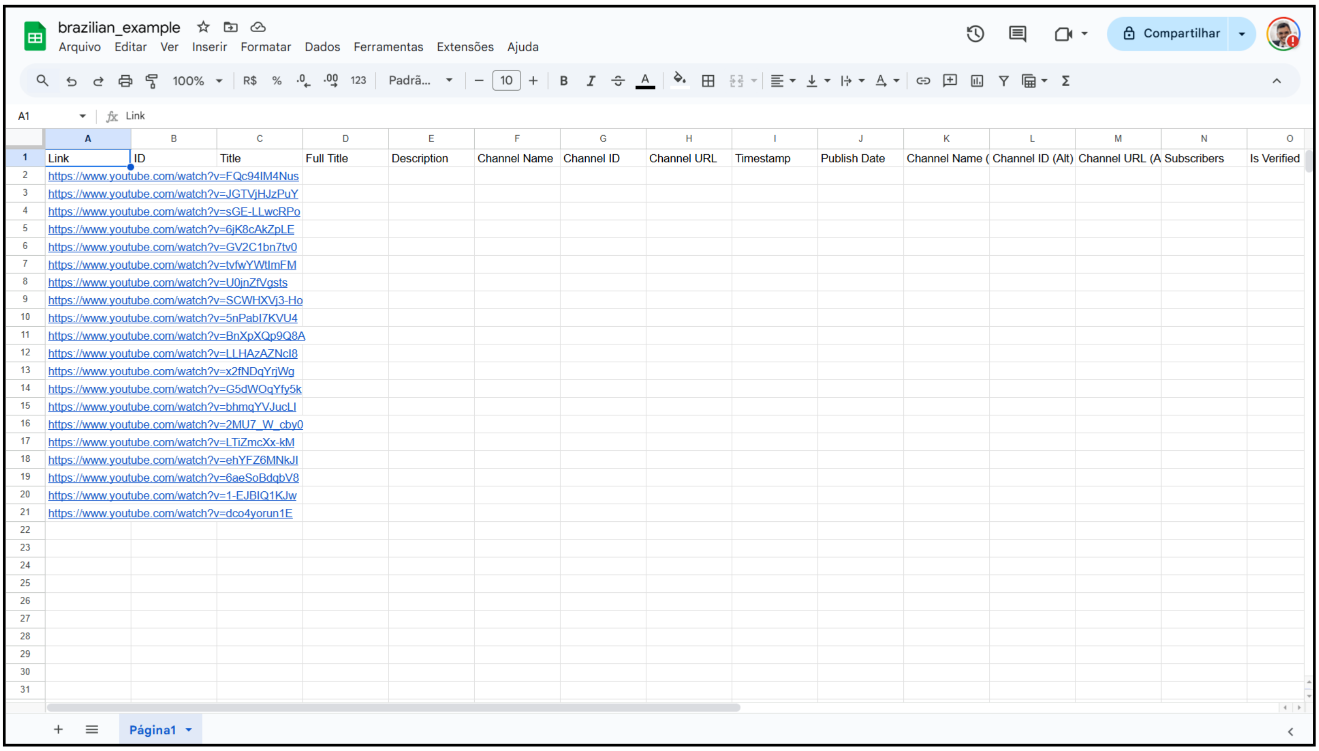
end_date: 31/12/2024

Mostrar código

```
Searching YouTube: 100%  2/2 [01:07<00:00, 33.75s/it]
Processing "presidente lula' -bolsonaro -eleição':  10/? [00:35<00:00,  3.55s/it]
Processing 'election usa -biden -kamala -trump':  10/? [00:32<00:00,  3.23s/it]
```

Source: Authors (2024)

Figure 4 - Result after running the second cell of the code



brazilian_example
Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda
Compartilhar
100%
R$ % .0 .00 123 Padrã... 10
A1 fx Link
Página1

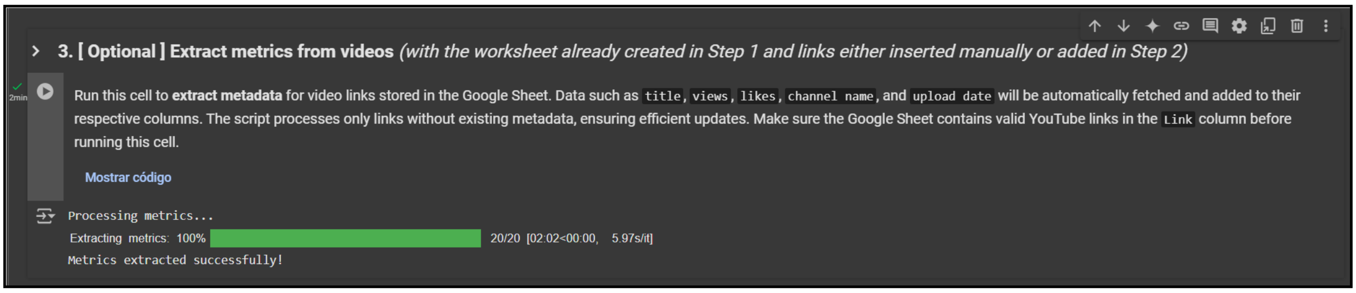
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Link	ID	Title	Full Title	Description	Channel Name	Channel ID	Channel URL	Timestamp	Publish Date	Channel Name (	Channel ID (Alt)	Channel URL (A	Subscribers	Is Verified
2	https://www.youtube.com/watch?v=FQc94IM4Nus														
3	https://www.youtube.com/watch?v=JGTVjHJzPuY														
4	https://www.youtube.com/watch?v=sGE-LLwcRPo														
5	https://www.youtube.com/watch?v=6jK8cAkZpLE														
6	https://www.youtube.com/watch?v=GV2C1bn7tv0														
7	https://www.youtube.com/watch?v=tvfwYWtImFM														
8	https://www.youtube.com/watch?v=U0jnZfVgsts														
9	https://www.youtube.com/watch?v=SCWHXVj3-Ho														
10	https://www.youtube.com/watch?v=5nPabI7KVU4														
11	https://www.youtube.com/watch?v=BnXpXQp9Q8A														
12	https://www.youtube.com/watch?v=LLHAzAZNcI8														
13	https://www.youtube.com/watch?v=x2fNDqYrjWg														
14	https://www.youtube.com/watch?v=G5dWOqYfy5k														
15	https://www.youtube.com/watch?v=bhmqYVJucLI														
16	https://www.youtube.com/watch?v=2MU7_W_cby0														
17	https://www.youtube.com/watch?v=LTiZmcXx-kM														
18	https://www.youtube.com/watch?v=ehYFZ6MNkJI														
19	https://www.youtube.com/watch?v=6aeSoBdqbV8														
20	https://www.youtube.com/watch?v=1-EJBIQ1KJw														
21	https://www.youtube.com/watch?v=dco4yorun1E														

Source: Authors (2024)

III. [Optional] Extract metrics from videos (with the worksheet already created in Step 1 and links either inserted manually or added in Step 2)

The code will now extract significant Metadata from each collected link. These are: [“ID”, “Title”, “Full Title”, “Description”, “Channel Name”, “Channel ID”, “Channel URL”, “Timestamp”, “Publish Date”, “Channel Name (Alt)”, “Channel ID (Alt)”, “Channel URL (Alt)”, “Subscribers”, “Is Verified”, “Location”, “Length (s)”, “Views”, “Likes”, “Dislikes”, “Reposts”, “Comments”, “Tags”, “Thumbnail”], ensuring that all relevant information is easily accessible in a structured and organized format.

Figure 5 - Third cell where Metadata is extracted



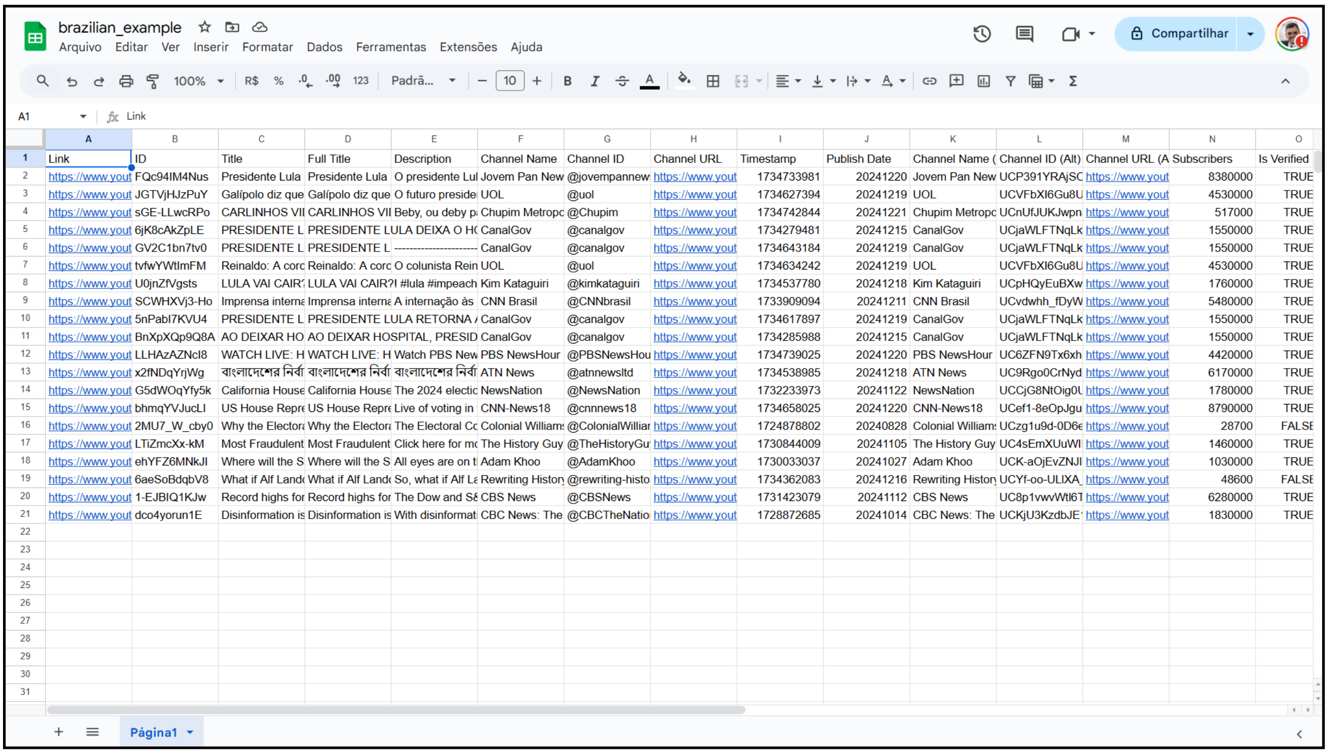
3. [Optional] Extract metrics from videos *(with the worksheet already created in Step 1 and links either inserted manually or added in Step 2)*

Run this cell to **extract metadata** for video links stored in the Google Sheet. Data such as `title`, `views`, `likes`, `channel name`, and `upload date` will be automatically fetched and added to their respective columns. The script processes only links without existing metadata, ensuring efficient updates. Make sure the Google Sheet contains valid YouTube links in the `Link` column before running this cell.

```
Processing metrics...
Extracting metrics: 100% ████████████████ 20/20 [02:02<00:00,  5.97s/it]
Metrics extracted successfully!
```

Source: Authors (2024)

Figure 6 - Result after running the third cell of the code



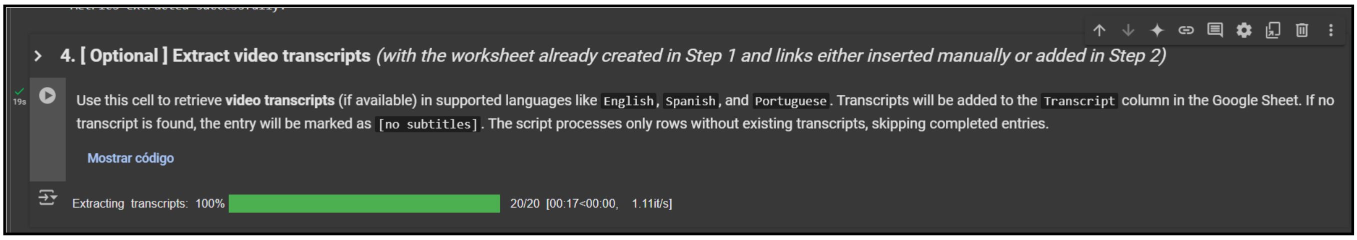
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Link	ID	Title	Full Title	Description	Channel Name	Channel ID	Channel URL	Timestamp	Publish Date	Channel Name (	Channel ID (Alt)	Channel URL (A	Subscribers	Is Verified
2	https://www.yout	FQc94IM4Nus	Presidente Lula	Presidente Lula	O presidente Lul	Jovem Pan New	@jovempannew	https://www.yout	1734733981	20241220	Jovem Pan New	UCP391YRAjSC	https://www.yout	8380000	TRUE
3	https://www.yout	JGTVjHJzPuY	Galípolo diz que	Galípolo diz que	O futuro preside	UOL	@uol	https://www.yout	1734627394	20241219	UOL	UCVFbXI6Gu8U	https://www.yout	4530000	TRUE
4	https://www.yout	sGE-LLwcRPo	CARLINHOS VII	CARLINHOS VII	Beby, ou deby pa	Chupim Metropo	@Chupim	https://www.yout	1734742844	20241221	Chupim Metropo	UCnUfJUKJwpn	https://www.yout	517000	TRUE
5	https://www.yout	6jK8cAkZpLE	PRESIDENTE L	PRESIDENTE LULA DEIXA O H(		CanalGov	@canalgov	https://www.yout	1734279481	20241215	CanalGov	UCjaWLFTNqLk	https://www.yout	1550000	TRUE
6	https://www.yout	GV2C1bn7tv0	PRESIDENTE L	PRESIDENTE L	--------------------	CanalGov	@canalgov	https://www.yout	1734643184	20241219	CanalGov	UCjaWLFTNqLk	https://www.yout	1550000	TRUE
7	https://www.yout	tvfwYWtlmFM	Reinaldo: A corc	Reinaldo: A corc	O colunista Rein	UOL	@uol	https://www.yout	1734634242	20241219	UOL	UCVFbXI6Gu8U	https://www.yout	4530000	TRUE
8	https://www.yout	U0jnZfVgsts	LULA VAI CAIR?	LULA VAI CAIR?! #lula #impeach		Kim Kataguiri	@kimkataguiri	https://www.yout	1734537780	20241218	Kim Kataguiri	UCpHQyEuBXw	https://www.yout	1760000	TRUE
9	https://www.yout	SCWHXVj3-Ho	Imprensa interna	Imprensa interna	A internação às	CNN Brasil	@CNNbrasil	https://www.yout	1733909094	20241211	CNN Brasil	UCvdwhh_fDyW	https://www.yout	5480000	TRUE
10	https://www.yout	5nPabI7KVU4	PRESIDENTE L	PRESIDENTE LULA RETORNA /		CanalGov	@canalgov	https://www.yout	1734617897	20241219	CanalGov	UCjaWLFTNqLk	https://www.yout	1550000	TRUE
11	https://www.yout	BnXpXQp9Q8A	AO DEIXAR HO	AO DEIXAR HOSPITAL, PRESID		CanalGov	@canalgov	https://www.yout	1734285988	20241215	CanalGov	UCjaWLFTNqLk	https://www.yout	1550000	TRUE
12	https://www.yout	LLHAzAZNcI8	WATCH LIVE: H	WATCH LIVE: H	Watch PBS New	PBS NewsHour	@PBSNewsHou	https://www.yout	1734739025	20241220	PBS NewsHour	UC6ZFN9Tx6xh	https://www.yout	4420000	TRUE
13	https://www.yout	x2fNDqYrjWg	বাংলাদেশের নির্বা	বাংলাদেশের নির্বা	বাংলাদেশের নির্বা	ATN News	@atnnewsltd	https://www.yout	1734538985	20241218	ATN News	UC9Rgo0CrNyd	https://www.yout	6170000	TRUE
14	https://www.yout	G5dWOqYfy5k	California House	California House	The 2024 electic	NewsNation	@NewsNation	https://www.yout	1732233973	20241122	NewsNation	UCCjG8NtOig0U	https://www.yout	1780000	TRUE
15	https://www.yout	bhmqYVJucLI	US House Repre	US House Repre	Live of voting in	CNN-News18	@cnnnews18	https://www.yout	1734658025	20241220	CNN-News18	UCef1-8eOpJgu	https://www.yout	8790000	TRUE
16	https://www.yout	2MU7_W_cby0	Why the Electora	Why the Electora	The Electoral Co	Colonial Williams	@ColonialWilliar	https://www.yout	1724878802	20240828	Colonial Williams	UCzg1u9d-0D6e	https://www.yout	28700	FALSE
17	https://www.yout	LTiZmcXx-kM	Most Fraudulent	Most Fraudulent	Click here for mo	The History Guy	@TheHistoryGu	https://www.yout	1730844009	20241105	The History Guy	UC4sEmXUuWI	https://www.yout	1460000	TRUE
18	https://www.yout	ehYFZ6MNkJI	Where will the S	Where will the S	All eyes are on t	Adam Khoo	@AdamKhoo	https://www.yout	1730033037	20241027	Adam Khoo	UCK-aOjEvZNJI	https://www.yout	1030000	TRUE
19	https://www.yout	6aeSoBdqbV8	What if Alf Lando	What if Alf Lando	So, what if Alf La	Rewriting History	@rewriting-histo	https://www.yout	1734362083	20241216	Rewriting History	UCYf-oo-ULIXA_	https://www.yout	48600	FALSE
20	https://www.yout	1-EJBIQ1KJw	Record highs for	Record highs for	The Dow and S&	CBS News	@CBSNews	https://www.yout	1731423079	20241112	CBS News	UC8p1vwvWtl6T	https://www.yout	6280000	TRUE
21	https://www.yout	dco4yorun1E	Disinformation is	Disinformation is	With disinformati	CBC News: The	@CBCTheNatio	https://www.yout	1728872685	20241014	CBC News: The	UCKjU3KzdbJE	https://www.yout	1830000	TRUE
22															
23															
24															
25															
26															
27															
28															
29															
30															
31															

Source: Authors (2024)

IV. [Optional] Extract video transcripts (with the worksheet already created in Step 1 and links either inserted manually or added in Step 2)

If subtitles are available, the tool fetches the transcript in multiple languages and stores it in the sheet. Videos without subtitles are marked accordingly.

Figure 7 - The last cell to be evaluated, where Transcriptions are extracted, if available



4. [Optional] Extract video transcripts *(with the worksheet already created in Step 1 and links either inserted manually or added in Step 2)*

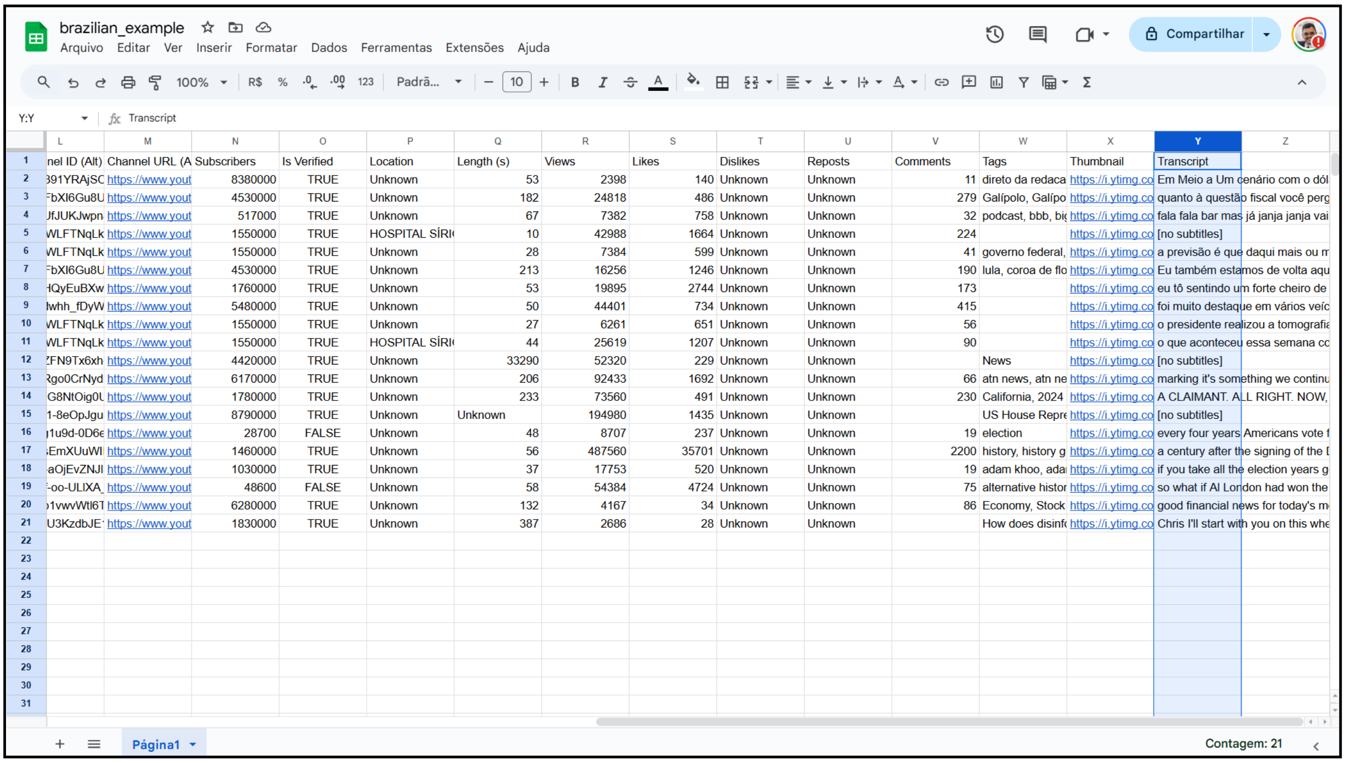
Use this cell to retrieve **video transcripts** (if available) in supported languages like `English`, `Spanish`, and `Portuguese`. Transcripts will be added to the `Transcript` column in the Google Sheet. If no transcript is found, the entry will be marked as `[no subtitles]`. The script processes only rows without existing transcripts, skipping completed entries.

Mostrar código

```
Extracting transcripts: 100% 20/20 [00:17<00:00,  1.11it/s]
```

Source: Authors (2024)

Figure 8 - Result after running the last cell of the code



brazilian_example
Arquivo Editar Ver Inserir Formatar Dados Ferramentas Extensões Ajuda
Compartilhar
100%
Padrã...
10
Página1
Contagem: 21

Y:Y *fx* Transcript

	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	nel ID (Alt)	Channel URL (A	Subscribers	Is Verified	Location	Length (s)	Views	Likes	Dislikes	Reposts	Comments	Tags	Thumbnail	Transcript
2	391YRAjSC	https://www.yout	8380000	TRUE	Unknown	53	2398	140	Unknown	Unknown	11	direto da redaca	https://i.ytimg.co	Em Meio a Um cenário com o dól
3	FbXI6Gu8U	https://www.yout	4530000	TRUE	Unknown	182	24818	486	Unknown	Unknown	279	Galípolo, Galípo	https://i.ytimg.co	quanto à questão fiscal você perg
4	JfJUKJwpn	https://www.yout	517000	TRUE	Unknown	67	7382	758	Unknown	Unknown	32	podcast, bbb, bi	https://i.ytimg.co	fala fala bar mas já janja janja vai
5	WLFTNqLk	https://www.yout	1550000	TRUE	HOSPITAL SÍRI	10	42988	1664	Unknown	Unknown	224		https://i.ytimg.co	[no subtitles]
6	WLFTNqLk	https://www.yout	1550000	TRUE	Unknown	28	7384	599	Unknown	Unknown	41	governo federal,	https://i.ytimg.co	a previsão é que daqui mais ou m
7	FbXI6Gu8U	https://www.yout	4530000	TRUE	Unknown	213	16256	1246	Unknown	Unknown	190	lula, coroa de flo	https://i.ytimg.co	Eu também estamos de volta aqu
8	HQyEuBXw	https://www.yout	1760000	TRUE	Unknown	53	19895	2744	Unknown	Unknown	173		https://i.ytimg.co	eu tô sentindo um forte cheiro de
9	lwhh_fDyW	https://www.yout	5480000	TRUE	Unknown	50	44401	734	Unknown	Unknown	415		https://i.ytimg.co	foi muito destaque em vários veíc
10	WLFTNqLk	https://www.yout	1550000	TRUE	Unknown	27	6261	651	Unknown	Unknown	56		https://i.ytimg.co	o presidente realizou a tomografi
11	WLFTNqLk	https://www.yout	1550000	TRUE	HOSPITAL SÍRI	44	25619	1207	Unknown	Unknown	90		https://i.ytimg.co	o que aconteceu essa semana co
12	ZFN9Tx6xh	https://www.yout	4420000	TRUE	Unknown	33290	52320	229	Unknown	Unknown		News	https://i.ytimg.co	[no subtitles]
13	Rgo0CrNyd	https://www.yout	6170000	TRUE	Unknown	206	92433	1692	Unknown	Unknown	66	atn news, atn ne	https://i.ytimg.co	marking it's something we continu
14	G8NtOig0U	https://www.yout	1780000	TRUE	Unknown	233	73560	491	Unknown	Unknown	230	California, 2024	https://i.ytimg.co	A CLAIMANT. ALL RIGHT. NOW,
15	1-8eOpJgu	https://www.yout	8790000	TRUE	Unknown	Unknown	194980	1435	Unknown	Unknown		US House Repre	https://i.ytimg.co	[no subtitles]
16	g1u9d-0D6e	https://www.yout	28700	FALSE	Unknown	48	8707	237	Unknown	Unknown	19	election	https://i.ytimg.co	every four years Americans vote f
17	sEmXUuWI	https://www.yout	1460000	TRUE	Unknown	56	487560	35701	Unknown	Unknown	2200	history, history g	https://i.ytimg.co	a century after the signing of the [
18	aOjEvZNJI	https://www.yout	1030000	TRUE	Unknown	37	17753	520	Unknown	Unknown	19	adam khoo, ada	https://i.ytimg.co	if you take all the election years g
19	-oo-ULIXA_	https://www.yout	48600	FALSE	Unknown	58	54384	4724	Unknown	Unknown	75	alternative histor	https://i.ytimg.co	so what if Al London had won the
20	01vwvWtl6T	https://www.yout	6280000	TRUE	Unknown	132	4167	34	Unknown	Unknown	86	Economy, Stock	https://i.ytimg.co	good financial news for today's m
21	U3KzdbJE	https://www.yout	1830000	TRUE	Unknown	387	2686	28	Unknown	Unknown		How does disinf	https://i.ytimg.co	Chris I'll start with you on this whe

Source: Authors (2024)

CODE

I. Create or load a Google Sheet

Table 1 - First cell approaches and code

Approach description	Code description

This cell sets up the functionality for creating or accessing a Google Sheet, which acts as the storage location for the extracted YouTube data. First, the required Python libraries are installed to ensure the necessary tools for interacting with YouTube and Google Sheets are available. !pip install installs the required libraries (yt-dlp, youtube-transcript-api, gspread, tqdm, and scrape-youtube) for video data extraction and Google Sheets interaction. Additionally, auth and gspread are used for Google authentication and worksheet management, and google.auth enables default authentication for accessing Google Sheets. Then, the code authenticates the user's Google account, granting the tool permission to access and modify Google Sheets. This is done through auth.authenticate_user(), which prompts the user to login and grant access to their Google account, enabling the tool to interact with Google Sheets. Through creds, _ = default(), the user's credentials are obtained for authentication, and gc = gspread.authorize(creds) authorizes access to Google Sheets using the authenticated credentials. A predefined list of column names to organize the extracted data (e.g., video links, titles, descriptions, channel information, views, likes, and transcripts) is created through COLUMN_HEADERS. These headers ensure the worksheet is formatted for consistent data entry, and the load_or_create_sheet function manages the creation or retrieval of the Google Sheet. It attempts to open an existing Google Sheet with the name specified in sheet_name, and, If the specified sheet does not exist, it creates a new Google Sheet.	!pip install -q yt-dlp youtube-transcript-api gspread tqdm scrape-youtube
	
	sheet_name = "name_your_worksheet" # @param {type:"string"}
	from google.colab import auth
	import gspread
	from google.auth import default
	
	auth.authenticate_user()
	creds, _ = default()
	gc = gspread.authorize(creds)
	
	COLUMN_HEADERS = [
	    "Link", "ID", "Title", "Full Title", "Description", "Channel Name", "Channel ID",
	    "Channel URL", "Timestamp", "Publish Date", "Channel Name (Alt)", "Channel ID (Alt)",
	    "Channel URL (Alt)", "Subscribers", "Is Verified", "Location", "Length (s)",
	    "Views", "Likes", "Dislikes", "Reposts", "Comments", "Tags", "Thumbnail", "Transcript"
	]
	
	def load_or_create_sheet(sheet_name, headers):
	    try:
	        sheet = gc.open(sheet_name)
	    except gspread.SpreadsheetNotFound:
	        sheet = gc.create(sheet_name)
	        print(f"Worksheet '{sheet_name}' created successfully.")
	    worksheet = sheet.get_worksheet(0) or sheet.add_worksheet(title="Sheet1", rows="100", cols="26")
	    current_headers = worksheet.row_values(1)
	    if current_headers != headers:
	        worksheet.insert_row(headers, 1)
	    print(f"Access your worksheet here: https://docs.google.com/spreadsheets/d/{sheet.id}/edit")
	    return worksheet
	
	worksheet = load_or_create_sheet(sheet_name, COLUMN_HEADERS)

Source: Authors (2024)

II. Search YouTube videos and save links

Table 2 - Second cell approaches and code

<table>
<tr><th>Approach description</th><th>Code description</th></tr>
<tr><td>This cell sets up the functionality for searching YouTube videos and saving their links based on specified criteria. First, it defines the input parameters for user queries, including keywords, maximum results, and a date range. You may input up to five queries (query_1 to query_5), where each query can include positive and negative keywords to refine the search. The max_results parameter, controlled via a slider, specifies the maximum number of results to retrieve for each query, while the start_date and end_date parameters enable filtering of videos based on their upload dates. This is the point where libraries are leveraged to process the queries effectively. The scrapetube library retrieves search results from YouTube based on the defined queries, while yt_dlp handles metadata extraction for the retrieved video links. The datetime module is used for parsing and comparing dates, and tqdm provides a progress bar for visual feedback, ensuring that users can monitor the status of the operation. The input queries are collected into a list and filtered to remove any empty entries. Dates are converted from strings into datetime objects to allow proper comparison and filtering during the video search process. This ensures that the tool processes only the videos that meet the defined criteria. Two helper functions are also defined to streamline the process. The get_video_details (video_id) function uses yt_dlp to extract basic video metadata such as the title, upload date, and video link. The video_matches_date (video, date_start, date_end) function checks if a video's upload date falls within the specified date range, ensuring that only relevant videos are processed further. The main function, search_youtube_to_sheet, performs the core task of searching YouTube and saving the results. It iterates through each query, retrieves video results using scrapetube.get_search(query), and processes each video to extract its ID and metadata using get_video_details. It then verifies if the video’s upload date matches the specified range using video_matches_date. If the video meets the criteria, its link is saved into the Google Sheet at the next available row. Finally, the function is executed to process all user-defined queries, with progress displayed for each query using tqdm.</td><td><pre>
query_1 = "'presidente lula' -bolsonaro -eleição" # @param {type:"string"}
query_2 = "election usa -biden -kamala -trump" # @param {type:"string"}
query_3 = "" # @param {type:"string"}
query_4 = "" # @param {type:"string"}
query_5 = "" # @param {type:"string"}
max_results = 10 # @param {type:"slider", min:1, max:1000, step:1}
start_date = '2020-01-01' # @param {type:"date"}
end_date = '2024-12-31' # @param {type:"date"}

import scrapetube
from yt_dlp import YoutubeDL
from datetime import datetime
from tqdm.notebook import tqdm

queries = [query_1, query_2, query_3, query_4, query_5]
queries = [q for q in queries if q.strip()] # Remove empty queries

date_start = datetime.strptime(start_date, "%Y-%m-%d")
date_end = datetime.strptime(end_date, "%Y-%m-%d")

def get_video_details(video_id):
    url = f"https://www.youtube.com/watch?v={video_id}"
    ydl_opts = {"quiet": True}
    with YoutubeDL(ydl_opts) as ydl:
        info = ydl.extract_info(url, download=False)
        return {
            "title": info.get("title", "Unknown"),
            "upload_date": info.get("upload_date", "00000000"),
            "link": url,
        }

def video_matches_date(video, date_start, date_end):
    try:
        upload_date = datetime.strptime(video["upload_date"], "%Y%m%d")
        return date_start <= upload_date <= date_end
    except:
        return False

def search_youtube_to_sheet(worksheet, queries, max_results, date_start,
date_end):
    next_row = len(worksheet.col_values(1)) + 1
    for query in tqdm(queries, desc="Searching YouTube"):
        videos = scrapetube.get_search(query)
        results = []
        for video in tqdm(videos, desc=f"Processing '{query}'", leave=False):
            if len(results) >= max_results:
                break
            video_id = video["videoId"]
            video_details = get_video_details(video_id)
            if video_matches_date(video_details, date_start, date_end):
                results.append(video_details)
                worksheet.update_cell(next_row, 1, video_details["link"])
                next_row += 1

search_youtube_to_sheet(worksheet, queries, max_results, date_start, date_end)
</pre></td></tr>
</table>

Source: Authors (2024)

III. Extract metrics from videos

Table 03 - Third cell approaches and code

<table>
<tr><th>Approach description</th><th>Code description</th></tr>
<tr><td>This cell is responsible for extracting detailed video metadata (metrics) from the links previously saved in the Google Sheet. It iterates through each video link, retrieves relevant data using the yt_dlp library, and updates the corresponding rows in the Google Sheet with the extracted information. The process begins with importing the required libraries that together form the backbone of the video data extraction functionability: the aforementioned yt_dlp is used to fetch comprehensive metadata for each video, re allows the use of regular expressions to parse video IDs from YouTube URLs, and tqdm provides a progress bar for real-time visual feedback during the operation. The extract_video_metrics (worksheet) function starts by scanning the Google Sheet for video links that need processing by identifying rows where a link exists in the first column, but the second column (Video ID) is empty. These links are added to a list for further processing, ensuring that previously processed links are skipped to save time and avoid duplication. For each video link in the list, the tool extracts the video ID using the extract_video_id() function and fetches the associated metadata using yt_dlp. This metadata includes essential video details such as title, full title, description, duration, and tags. It also retrieves channel information like the channel name, ID, URL, number of subscribers, and verification status. Performance metrics, including views, likes, dislikes, reposts, and comment count, are recorded as well. Additionally, the upload date, timestamp, location, and thumbnail URL are extracted. Tags are converted into a comma-separated string for easier analysis and storage. The extracted metadata is then systematically added to the Google Sheet. Each piece of information is inserted into its corresponding column based on the predefined headers, ensuring that all data is well-organized and ready for analysis. This step provides users with structured insights directly in the worksheet. The function also includes robust error handling: if an issue arises while processing a video link, such as an invalid link or network error, the tool logs the error and moves on to the next link, ensuring it that the process continues uninterrupted for the remaining videos. Finally, the function is executed, and a progress bar is displayed using tqdm, indicating how many videos have been processed and how many remain. Upon completion, the tool prints a success message, confirming that all video metrics have been extracted and stored in the Google Sheet. This structured and automated process simplifies the collection and organization of comprehensive video data for analysis.</td><td><pre>
from yt_dlp import YoutubeDL
import re
from tqdm.notebook import tqdm

def extract_video_id(link):
    """
    Extracts the video ID from a YouTube link.

    Args:
        link (str): Full video URL.

    Returns:
        str: Extracted video ID.
    """
    match = re.search(r"v=([^&]+)", link)
    return match.group(1) if match else "Unknown"

def extract_video_metrics(worksheet):
    """
    Extracts video metrics and updates the worksheet.

    Args:
        worksheet: Google Worksheet object.
    """
    ydl_opts = {
        'quiet': True,
        'no_warnings': True,
        'skip_download': True,
    }

    # Collect all links with missing metadata
    links = []
    i = 2
    while worksheet.cell(i, 1).value:
        if not worksheet.cell(i, 2).value:
            links.append((i, worksheet.cell(i, 1).value))
        i += 1

    # Iterate over links and process each video
    for index, link in tqdm(links, desc="Extracting metrics"):
        try:
            video_id = extract_video_id(link) # Extract video ID from the link

            with YoutubeDL(ydl_opts) as ydl:
                info = ydl.extract_info(link, download=False)

            tags = info.get('tags', [])
            tags_str = ', '.join(tags) if isinstance(tags, list) else 'Unknown'

            # Update worksheet with extracted data
            worksheet.update_cell(index, 2, video_id) # Video ID
            worksheet.update_cell(index, 3, info.get('title', 'Unknown'))
            worksheet.update_cell(index, 4, info.get('fulltitle', 'Unknown'))
            worksheet.update_cell(index, 5, info.get('description', '').replace('\n', ' '))
            worksheet.update_cell(index, 6, info.get('uploader', 'Unknown'))
            worksheet.update_cell(index, 7, info.get('uploader_id', 'Unknown'))
            worksheet.update_cell(index, 8, info.get('uploader_url', 'Unknown'))
            worksheet.update_cell(index, 9, info.get('timestamp', 'Unknown'))
            worksheet.update_cell(index, 10, info.get('upload_date', 'Unknown'))
            worksheet.update_cell(index, 11, info.get('channel', 'Unknown'))
            worksheet.update_cell(index, 12, info.get('channel_id', 'Unknown'))
            worksheet.update_cell(index, 13, info.get('channel_url', 'Unknown'))
            worksheet.update_cell(index, 14, info.get('channel_follower_count', 'Unknown'))
            worksheet.update_cell(index, 15, info.get('channel_is_verified', 'FALSE'))
            worksheet.update_cell(index, 16, info.get('location', 'Unknown'))
            worksheet.update_cell(index, 17, info.get('duration', 'Unknown'))
            worksheet.update_cell(index, 18, info.get('view_count', 'Unknown'))
            worksheet.update_cell(index, 19, info.get('like_count', 'Unknown'))
            worksheet.update_cell(index, 20, info.get('dislike_count', 'Unknown'))
            worksheet.update_cell(index, 21, info.get('repost_count', 'Unknown'))
            worksheet.update_cell(index, 22, info.get('comment_count', 'Unknown'))
            worksheet.update_cell(index, 23, tags_str)
            worksheet.update_cell(index, 24, info.get('thumbnail', 'Unknown'))
        except Exception as e:
            print(f"Error processing link {link}: {e}")

print("Processing metrics...")
extract_video_metrics(worksheet)
print("Metrics extracted successfully!")
</pre></td></tr>
</table>

Source: Authors (2024)

IV. Extract video transcripts

Table 04 - Fourth cell approaches and code

Approach description	Code description

YouTubeTranscriptAPI is used to fetch video transcripts, re enables the parsing of video IDs from the YouTube URLs, and tqdm, once more, provides a progress bar to monitor the process. The main function, extract_video_transcripts (worksheet), starts by collecting video links from the first column of the Google Sheet. It identifies rows where a video link is present, but the corresponding cell in the transcript column (column 25) is empty. It is configured to retrieve subtitles in multiple languages, specifically English (en), Spanish (es), and Portuguese (pt) as standard. This cleaned transcript is then inserted into the corresponding cell in the Google Sheet. If no transcript is available for a particular video (e.g., subtitles are disabled or unavailable), the tool updates the cell with the placeholder text "[no subtitles]". Finally, the function is executed, and the tqdm progress bar provides real-time feedback on the status of the operation. Users can track how many videos have been processed and how many re-	from youtube_transcript_api import YouTubeTranscriptApi
	import re
	from tqdm.notebook import tqdm
	
	def extract_video_transcripts(worksheet):
	    col_url = worksheet.col_values(1)
	    urls_to_process = [(idx, url) for idx, url in enumerate (col_url[1:], start=2) if not worksheet.cell(idx, 25).value]
	
	    for idx, url in tqdm(urls_to_process, desc="Extracting transcripts"):
	        video_id = re.search(r"v=([^&]+)", url)
	        video_id = video_id.group(1) if video_id else None
	        if video_id:
	            try:
	                transcript = YouTubeTranscriptApi.get_transcript(video_id, languages=['en', 'es', 'pt'])
	                transcript_text = ' '.join([item['text'].replace ('\n', ' ') for item in transcript])
	                worksheet.update_cell(idx, 25, transcript_text)
	            except Exception:
	                worksheet.update_cell(idx, 25, "[no subtitles]")

Source: Authors (2024)

CONCLUSIONS

The **YouTubeScrap** tool proposes a step forward in democratizing access to Social Media data, addressing the current challenges researchers and analysts face in collecting and analyzing online content. Designed as an open-source and freely available solution, the code's placement within the Google Colab environment ensures ease of use and accessibility for users at all levels of technical expertise as colab's cloud-based infrastructure eliminates the need for local installations and provides a ready-to-use environment, allowing users to focus on their research without worrying about software compatibility or setup complexities.

A standout feature of this tool is its ability to operate without requiring API keys. As access to online data becomes increasingly limited due to stricter API regulations and usage restrictions, **YouTubeScrap** provides a key-free solution that enables users to retrieve, analyze, and organize video data with ease. By eliminating the need for API keys, the tool removes barriers for those who may face challenges in obtaining credentials, promoting equitable access to critical data. Additionally, **YouTubeScrap** has been designed to align with international data privacy regulations, ensuring that its operation respects the principles of ethical data handling while empowering users to engage in compliant research and analysis.

Beyond its technical capabilities, this tool was developed with a clear mission: to support and enable research, media analysis, and public discourse studies in a time when data availability is diminishing. By automating the extraction of video links, metadata, and transcripts, it not only saves time but also standardizes the organization of large datasets, making them more actionable. **YoutubeScrap** bridges the gap between data inaccessibility and the growing need for systematic analysis of digital platforms: API-free design and integration with widely accessible technologies like Google Sheets highlight its potential to support academic, media, and communication studies while fostering innovation in research.

As a reflection of the value of open-source development in advancing knowledge, this tool is a critical resource for empowering researchers and practitioners in the field of Computational Social Sciences. While the evolving landscape of data accessibility presents ongoing challenges, we hope that **YouTubeScrap** remains operational long enough.

Its use is highly encouraged and recommended for academic and scientific research, content analysis, sentiment analysis, and speech analysis. While it is free to use and modify, the responsibility for its use and any modifications lies with the user. Feel free to explore, utilize, and adapt the code to suit your needs, but please ensure you comply with YouTube's terms of service and data privacy regulations. This tool is released under a free and open-source license. When using or modifying the tool, please ensure to provide appropriate credit and citation. Referencing the tool in your research is appreciated and contributes to its continued development and improvement.

REFERENCES

Silva, Ergon Cugler de Moraes; Rocha, Isabela. **YouTubeScrap: A comprehensive tool for scraping YouTube data and transcript.** (Dec, 2024). Available at: https://github.com/ergoncugler/web-scraping-youtube.

Statista. **Leading countries based on YouTube audience size as of July 2024 (in millions)** (2024). Available at: https://www.statista.com/statistics/280685/number-of-monthly-unique-youtube-users/.

Statista. **Leading social media platforms in Brazil 2023, by reach** (2023). Available at: https://www.statista.com/statistics/1307747/social-networks-penetration-brazil/.

NOTES

[1] See Code available on GitHub: https://github.com/ergoncugler/web-scraping-youtube

ISSN 2357-7975