

Lecture 02

Jupyter Notebook - YouTube Transcript API

Marcel Turcotte

2024-09-01

Learning objective

- **Illustrate** the process of identifying and resolving missing library issues in Google Colab.

! Important

This example is meant to be executed in [Google Colab](#).

YouTube Transcript API

In this notebook, we aim to utilize the [YouTube Transcript API](#) to automatically download the transcript of the video titled [Can Machines Think?](#) by Noam Chomsky.

First, let's import `YouTubeTranscriptApi` and `TextFormatter` from `youtube_transcript_api`.

```
from youtube_transcript_api import YouTubeTranscriptApi
from youtube_transcript_api.formatters import TextFormatter
```

Executing the code cell above will result in an error, as the `youtube_transcript_api` library is not installed by default in Google Colab.

```
-----
ModuleNotFoundError                                Traceback (most recent call last)
<ipython-input-1-c8308591d925> in <cell line: 2>()
      1 # ! pip install youtube-transcript-api
----> 2 from youtube_transcript_api import YouTubeTranscriptApi
      3 from youtube_transcript_api.formatters import TextFormatter
```

ModuleNotFoundError: No module named 'youtube_transcript_api'

NOTE: If your import is failing due to a missing package, you can manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the "Open Examples" button below.

This issue can be resolved by adding the following line of code before the first import statement. Try it!

```
! pip install youtube-transcript-api
```

Once this issue has been solved, we can download and print the transcript. Try it!

```
transcript = YouTubeTranscriptApi.get_transcript("Ex9GbzX6tMo")  
formatter = TextFormatter()  
input_text = formatter.format_transcript(transcript)  
print(input_text)
```

Exploration

! allows to run [Unix/Linux shell commands in IPython](#). Create a code cell and try these commands.

- ! `uname -a` displays information about the system.
- ! `ls` displays the content of the current directory.
- ! `ls /` displays the content of the root directory.
- ! `pwd` returns working directory name.

These commands are useful for debugging code, as they provide information about the computing environment, such as the operating system version and the contents of the local directory.