

whisperyoutube

August 14, 2024

#Creating YouTube transcripts with OpenAI's Whisper model

###**Note: For faster performance set your runtime to “GPU”** Click on “Runtime” in the menu and click “Change runtime type”. Select “GPU”.

Step 1. Follow the instructions in each block and select the options you want **Step 2.** Get the url of the video you want to transcribe **Step 3.** Refresh the folder on the left and download your transcript **Step 4.** Go to your YouTube account and upload the transcript to the video it came from and use “autosync.”

What is this? This is a Python notebook that creates a transcript from a YouTube url using OpenAI's Whisper transcription model that you can then upload to YouTube using the autosync feature to create captions.

What is OpenAI's Whisper model? Whisper is an automatic speech recognition (ASR) neural net created by OpenAI that transcribes audio at close to human level. **Why use this?** The quality of the OpenAI Whisper model is amazing (I am slightly biased, but seriously, check it out.) You can also use it to transcribe in other languages. **What do the different model sizes do?** Each model size has an improvement in quality – especially with different languages. I've found that for a YouTube video with clear speech, the base model works really well. If you see transcription errors, you can try a larger model. **Do I need timestamps?** Nope. YouTube's autosync function will match the text to the spoken words and syncs up really well. All you need is each spoken sentence in a .txt file. **How do I do this?** Just follow each step. If you've never used Colab or a Python notebook, don't panic. It's super easy and runs in the cloud. **Does this cost anything to use?** Nope. You can use Colab for free and Whisper is an open source model. [Tips for creating a YouTube transcript file](#) [Information on OpenAI's Whisper model](#) [OpenAI's Whisper GitHub page](#)

```
[ ]: """
1. Click the start button in the upper left side of this block to load the
   ↪necessary libraries

You will need to run this every time you reload this notebook.
"""

!pip install git+https://github.com/openai/whisper.git
!sudo apt update && sudo apt install ffmpeg
!pip install librosa
!pip install yt-dlp
```

```
import whisper
import time
import librosa
import re
import yt_dlp
```

Requirement already satisfied: youtube_dl in /usr/local/lib/python3.10/dist-packages (2021.12.17)

Collecting git+https://github.com/openai/whisper.git

Cloning https://github.com/openai/whisper.git to /tmp/pip-req-build-n2pg77w2

Running command git clone --filter=blob:none --quiet

https://github.com/openai/whisper.git /tmp/pip-req-build-n2pg77w2

Resolved https://github.com/openai/whisper.git to commit

ba3f3cd54b0e5b8ce1ab3de13e32122d0d5f98ab

Installing build dependencies ... done

Getting requirements to build wheel ... done

Preparing metadata (pyproject.toml) ... done

Requirement already satisfied: numba in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (0.60.0)

Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (1.26.4)

Requirement already satisfied: torch in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (2.3.1+cu121)

Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (4.66.5)

Requirement already satisfied: more-itertools in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (10.3.0)

Requirement already satisfied: tiktoken in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (0.7.0)

Requirement already satisfied: triton<3,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from openai-whisper==20231117) (2.3.1)

Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from triton<3,>=2.0.0->openai-whisper==20231117) (3.15.4)

Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.10/dist-packages (from numba->openai-whisper==20231117) (0.43.0)

Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.10/dist-packages (from tiktoken->openai-whisper==20231117) (2024.5.15)

Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.10/dist-packages (from tiktoken->openai-whisper==20231117) (2.32.3)

Requirement already satisfied: typing-extensions>=4.8.0 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (4.12.2)

Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages

(from torch->openai-whisper==20231117) (1.13.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (3.3)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (3.1.4)
Requirement already satisfied: fsspec in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (2024.6.1)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (12.1.105)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (12.1.105)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (12.1.105)
Requirement already satisfied: nvidia-cudnn-cu12==8.9.2.26 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (8.9.2.26)
Requirement already satisfied: nvidia-cublas-cu12==12.1.3.1 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (12.1.3.1)
Requirement already satisfied: nvidia-cufft-cu12==11.0.2.54 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (11.0.2.54)
Requirement already satisfied: nvidia-curand-cu12==10.3.2.106 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (10.3.2.106)
Requirement already satisfied: nvidia-cusolver-cu12==11.4.5.107 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (11.4.5.107)
Requirement already satisfied: nvidia-cuspars-cu12==12.1.0.106 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (12.1.0.106)
Requirement already satisfied: nvidia-nccl-cu12==2.20.5 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (2.20.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.1.105 in /usr/local/lib/python3.10/dist-packages (from torch->openai-whisper==20231117) (12.1.105)
Requirement already satisfied: nvidia-nvjitlink-cu12 in /usr/local/lib/python3.10/dist-packages (from nvidia-cusolver-cu12==11.4.5.107->torch->openai-whisper==20231117) (12.6.20)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken->openai-whisper==20231117) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.26.0->tiktoken->openai-whisper==20231117) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in
 /usr/local/lib/python3.10/dist-packages (from
 requests>=2.26.0->tiktoken->openai-whisper==20231117) (2.0.7)

Requirement already satisfied: certifi>=2017.4.17 in
 /usr/local/lib/python3.10/dist-packages (from
 requests>=2.26.0->tiktoken->openai-whisper==20231117) (2024.7.4)

Requirement already satisfied: MarkupSafe>=2.0 in
 /usr/local/lib/python3.10/dist-packages (from jinja2->torch->openai-
 whisper==20231117) (2.1.5)

Requirement already satisfied: mpmath<1.4,>=1.1.0 in
 /usr/local/lib/python3.10/dist-packages (from sympy->torch->openai-
 whisper==20231117) (1.3.0)

Hit:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease

Hit:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64
 InRelease

Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease

Ign:4 https://r2u.stat.illinois.edu/ubuntu jammy InRelease

Hit:5 https://r2u.stat.illinois.edu/ubuntu jammy Release

Hit:7 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease

Hit:8 http://archive.ubuntu.com/ubuntu jammy InRelease

Hit:9 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy
 InRelease

Hit:10 http://archive.ubuntu.com/ubuntu jammy-updates InRelease

Hit:11 http://archive.ubuntu.com/ubuntu jammy-backports InRelease

Hit:12 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

45 packages can be upgraded. Run 'apt list --upgradable' to see them.

W: Skipping acquire of configured file 'main/source/Sources' as
 repository 'https://r2u.stat.illinois.edu/ubuntu jammy InRelease' does not seem
 to provide it (sources.list entry misspelt?)

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

ffmpeg is already the newest version (7:4.4.2-0ubuntu0.22.04.1).

0 upgraded, 0 newly installed, 0 to remove and 45 not upgraded.

Requirement already satisfied: librosa in /usr/local/lib/python3.10/dist-
 packages (0.10.2.post1)

Requirement already satisfied: audioread>=2.1.9 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (3.0.1)

Requirement already satisfied: numpy!=1.22.0,!1.22.1,!1.22.2,>=1.20.3 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (1.26.4)

Requirement already satisfied: scipy>=1.2.0 in /usr/local/lib/python3.10/dist-
 packages (from librosa) (1.13.1)

Requirement already satisfied: scikit-learn>=0.20.0 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (1.3.2)

Requirement already satisfied: joblib>=0.14 in /usr/local/lib/python3.10/dist-

packages (from librosa) (1.4.2)
 Requirement already satisfied: decorator>=4.3.0 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (4.4.2)
 Requirement already satisfied: numba>=0.51.0 in /usr/local/lib/python3.10/dist-
 packages (from librosa) (0.60.0)
 Requirement already satisfied: soundfile>=0.12.1 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (0.12.1)
 Requirement already satisfied: pooch>=1.1 in /usr/local/lib/python3.10/dist-
 packages (from librosa) (1.8.2)
 Requirement already satisfied: soxr>=0.3.2 in /usr/local/lib/python3.10/dist-
 packages (from librosa) (0.4.0)
 Requirement already satisfied: typing-extensions>=4.1.1 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (4.12.2)
 Requirement already satisfied: lazy-loader>=0.1 in
 /usr/local/lib/python3.10/dist-packages (from librosa) (0.4)
 Requirement already satisfied: msgpack>=1.0 in /usr/local/lib/python3.10/dist-
 packages (from librosa) (1.0.8)
 Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
 packages (from lazy-loader>=0.1->librosa) (24.1)
 Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in
 /usr/local/lib/python3.10/dist-packages (from numba>=0.51.0->librosa) (0.43.0)
 Requirement already satisfied: platformdirs>=2.5.0 in
 /usr/local/lib/python3.10/dist-packages (from pooch>=1.1->librosa) (4.2.2)
 Requirement already satisfied: requests>=2.19.0 in
 /usr/local/lib/python3.10/dist-packages (from pooch>=1.1->librosa) (2.32.3)
 Requirement already satisfied: threadpoolctl>=2.0.0 in
 /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->librosa)
 (3.5.0)
 Requirement already satisfied: cffi>=1.0 in /usr/local/lib/python3.10/dist-
 packages (from soundfile>=0.12.1->librosa) (1.17.0)
 Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-
 packages (from cffi>=1.0->soundfile>=0.12.1->librosa) (2.22)
 Requirement already satisfied: charset-normalizer<4,>=2 in
 /usr/local/lib/python3.10/dist-packages (from
 requests>=2.19.0->pooch>=1.1->librosa) (3.3.2)
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-
 packages (from requests>=2.19.0->pooch>=1.1->librosa) (3.7)
 Requirement already satisfied: urllib3<3,>=1.21.1 in
 /usr/local/lib/python3.10/dist-packages (from
 requests>=2.19.0->pooch>=1.1->librosa) (2.0.7)
 Requirement already satisfied: certifi>=2017.4.17 in
 /usr/local/lib/python3.10/dist-packages (from
 requests>=2.19.0->pooch>=1.1->librosa) (2024.7.4)

```
[ ]: """
      2. Select the model you want to use.
```


Base works really well so it's the default.

(For multilingual, remove ".en" from the model name.)

Click the run button after you've made your choice (or left it at default.)
"""

```
# model = whisper.load_model("tiny.en")
model = whisper.load_model("base.en")
# model = whisper.load_model("small.en")
# model = whisper.load_model("medium.en")
# model = whisper.load_model("large")
```

[]: """
3. Click the run button and input your YouTube URL in the box below then click  enter.

The video will be loaded and the audio extracted (this is usually the longest  part of the process.)

Your transcript will appear in the folder on the left (you may have to refresh  the folder to see it.)

You can download the file when it's completed and upload it on your video's  detail page using "autosync."
"""

```
# This will prompt you for a YouTube video URL
url = input("Enter a YouTube video URL: ")

# Create a youtube-dl options dictionary
ydl_opts = {
    # Specify the format as bestaudio/best
    'format': 'bestaudio/best',
    # Specify the post-processor as ffmpeg to extract audio and convert to mp3
    'postprocessors': [{
        'key': 'FFmpegExtractAudio',
        'preferredcodec': 'mp3',
        'preferredquality': '192',
    }],
    # Specify the output filename as the video title
    'outtmpl': '%(title)s.%(ext)s',
}

# Download the video and extract the audio
with yt_dlp.YoutubeDL(ydl_opts) as ydl: # Use yt-dlp.YoutubeDL
    ydl.download([url])
```

```

# Get the path of the file
file_path = ydl.prepare_filename(ydl.extract_info(url, download=False))
file_path = file_path.replace('.webm', '.mp3')
file_path = file_path.replace('.m4a', '.mp3')

# Get the duration
duration = librosa.get_duration(filename=file_path)
start = time.time()
result = model.transcribe(file_path)
end = time.time()
seconds = end - start

print("Video length:", duration, "seconds")
print("Transcription time:", seconds)

# Split result["text"] on !,? and . , but save the punctuation
sentences = re.split("([!?.])", result["text"])

# Join the punctuation back to the sentences
sentences = ["".join(i) for i in zip(sentences[0::2], sentences[1::2])]
text = "\n\n".join(sentences)
for s in sentences:
    print(s)

# Save the file as .txt
name = "".join(file_path) + ".txt"
with open(name, "w") as f:
    f.write(text)

print("\n\n", "-"*100, "\n\nYour transcript is here:", name)

```

```

Enter a YouTube video URL: https://www.youtube.com/watch?v=P9cLmFGWfIw
[youtube] Extracting URL: https://www.youtube.com/watch?v=P9cLmFGWfIw
[youtube] P9cLmFGWfIw: Downloading webpage
[youtube] P9cLmFGWfIw: Downloading ios player API JSON
[youtube] P9cLmFGWfIw: Downloading web creator player API JSON
[youtube] P9cLmFGWfIw: Downloading player 410a4f15
[youtube] P9cLmFGWfIw: Downloading m3u8 information
[info] P9cLmFGWfIw: Downloading 1 format(s): 251
[download] Destination: Data Science in 4 Minutes Quick High Level
Overview.webm
[download] 100% of 2.66MiB in 00:00:00 at 3.05MiB/s
[ExtractAudio] Destination: Data Science in 4 Minutes Quick High Level
Overview.mp3
Deleting original file Data Science in 4 Minutes Quick High Level Overview.webm
(pass -k to keep)

```

```
[youtube] Extracting URL: https://www.youtube.com/watch?v=P9cLmFGWfIw
[youtube] P9cLmFGWfIw: Downloading webpage
[youtube] P9cLmFGWfIw: Downloading ios player API JSON
[youtube] P9cLmFGWfIw: Downloading web creator player API JSON
[youtube] P9cLmFGWfIw: Downloading m3u8 information
```

```
<ipython-input-10-0edae3c8832d>:46: FutureWarning: get_duration() keyword
argument 'filename' has been renamed to 'path' in version 0.10.0.
```

```
    This alias will be removed in version 1.0.
```

```
    duration = librosa.get_duration(filename=file_path)
```

```
Video length: 216.01525 seconds
```

```
Transcription time: 12.835594177246094
```

```
Hi, this is Jeff Heaton.
```

```
I'm going to tell you what data science is in four minutes, or at least try.
```

```
I better get going.
```

```
First up, data.
```

```
You can't have science without data.
```

```
Maybe you have a little, maybe you have a lot, but you've got to have data.
```

```
Often your data will be in a tabular form like this.
```

```
Think Microsoft Excel.
```

```
You've got columns.
```

```
You'd like to predict one of them.
```

```
Maybe you would like to predict the acceleration of a car based on these other
parameters.
```

```
You know the acceleration for a lot of these cars, but maybe there's some cars
where you don't know the acceleration.
```

```
You can train a model to predict that acceleration based on the ones that you
already know.
```

```
This is supervised learning.
```

```
If you're trying to predict a number, it's regression.
```

```
If you're trying to predict a class or a category or a type of car, it is
classification.
```

```
There's also unsupervised learning.
```

```
Maybe you don't know the acceleration for any car.
```

```
In this case, you take the values that you do have and try to cluster them.
```

```
Your unsupervised learning is going to look like this.
```

```
You're going to have clusters that those rows fall into.
```

```
Maybe you'll have the colors or maybe you won't, depending on the algorithm
that you're using.
```

```
It's probably up to you to assign which of those clusters new items most
closely aligned with.
```

```
Another important point is deciding how many clusters there are ahead of time.
```

```
Maybe you decide it.
```

```
Maybe the algorithm does.
```

```
It just depends on the algorithm.
```

```
Unsupervised and supervised learning are definitely the two biggest categories
in data science.
```

```
However, new categories are coming up all the time.
```


This is not the only way that this can be done.

Most of these algorithms, at least for supervised learning, are all about fitting a line to the actual data.

The actual data on the top one is the black line, which is very noisy usually.

The red line is the model that you're developing that as you train it gets closer and closer to the actual values, but you don't want it to be as jagged as the actual values, at least usually, or you will be in danger of overfitting.

Overfitting is one of your arch enemies as a data scientist.

You must prevent your models from overfitting.

There's other arch enemies as well, but it is one of the big ones.

Think of overfitting as like memorization.

If you're given a sample exam and you study just that sample exam and over and over and eventually you get 100% on the sample exam, will you pass the real exam?

Probably not.

There's underfitting too.

Underfitting occurs when the model type that you've chosen, such as an RBF, a Gaussian process, decision tree random forest, all these ones that you see across the very top of your screen, the columns are model types.

The rows are types of data.

You can see that the models themselves, the curved regions of those colors, they don't always fit the data just perfectly.

That is called bias error.

Your model simply cannot fit to the type of data that you have.

The solution to this is to use an ensemble where you use several columns to represent your data together.

But how do you get a better model of your data?

Well, there's many ways to do that, but one of the most common is something called feature engineering.

Feature engineering is where you create additional calculated columns in your data where maybe you take something like the acceleration and calculate that together with the size of the engine, maybe a ratio, to calculate an efficiency column that you add to this and pass it in with all your other data.

There you go.

Data science in four minutes.

Of course, there's a lot more to this field.

You can learn about that by subscribing to my YouTube channel.

Thank you for watching.

Thank you.

Your transcript is here: Data Science in 4 Minutes Quick High Level
Overview.mp3.txt