

How to create high-quality offline video transcriptions and subtitles using Whisper and Python

Olaf Janssen, Wikimedia coordinator of the KB national library of the Netherlands

Latest update: 5 November 2024

<https://doi.org/10.5281/zenodo.14047913>

I used to think that 'doing things with AI' was equivalent to smoking data centers, overheated servers, and massive cloud computing power. But this month, I had a jaw-dropping WTF OMG tech discovery: realizing that some AI tasks can run smoothly on a modest laptop, and even offline!

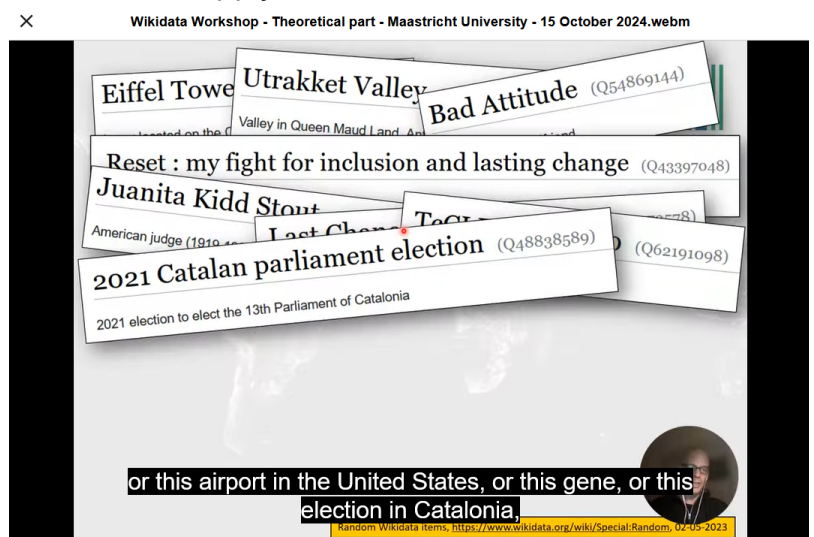
KB } national library
of the netherlands

I was searching for a solid solution to convert speech from a video file into text (also known as audio transcription, speech-to-text, or Automatic Speech Recognition, ASR) and found that this can all happen right on my own machine.

Why did I need audio transcriptions?

Using a recent video presentation I recorded, I wanted to apply ASR for several reasons:

- To **capture the full text** of my video, enabling ChatGPT to generate summaries, translations, blog posts, or social media content from it.
- To automatically **create subtitles**, enhancing accessibility for deaf and hard-of-hearing viewers and meeting the [WCAG](#) guideline to [provide captions for video content](#).
- And because it's **fun and educational** to explore new technology, especially when it turns out easier than expected, delivering quick, motivating, and useful results that encourage further experimentation.



Downsides of existing ASR services

Of course there are all kinds of existing audio-to-text cloud services, but they come with various downsides, including:

- Poor transcription quality, especially for names of things (so-called *named entities*, such as persons, places, organisations, journal titles etc.) and jargon words, which may need a lot of post-corrections;
- Limited number of supported languages;
- Privacy concerns: I want to avoid uploading my video to some sketchy AI transcription service, without knowing what will happen with it, especially when the source contains confidential content;
- Limited file sizes and/or video durations;
- Not wanting to publish your video on commercial platforms like YouTube, due to concerns about [public and open values](#), despite it offering good transcription and subtitle features;
- Costs, paid subscriptions etc.

For my little ASR project, I wanted to avoid these disadvantages as much as possible.

Whisper as a solution

As I work with ChatGPT regularly, I had heard of [Whisper, OpenAI's speech-to-text model](#), but I never actually looked into it or used it. So I thought I'd give it a try!

After some research to see if Whisper would suit my ASR needs, I found out that [this model excels in Dutch](#), but it also performs very well in English.

So that sounded very promising. But (as far as I know) Whisper doesn't offer a user-friendly front end, so I had to work with the API and Python. Fortunately, I found [this short blog post](#) to help me get started, and, combined with the [documentation](#), it was quite straightforward to set things up.

Later in this article, you'll see what I ultimately created with it, along with ready-to-use Python code so you can try it out for yourself.



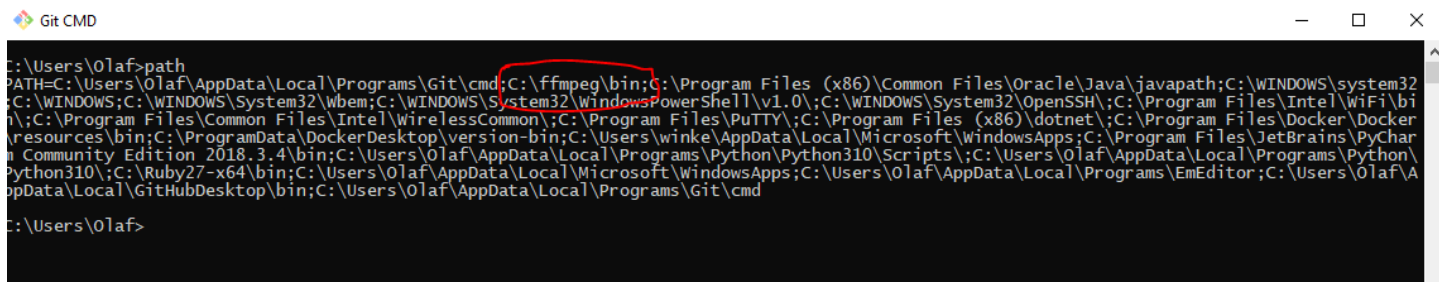
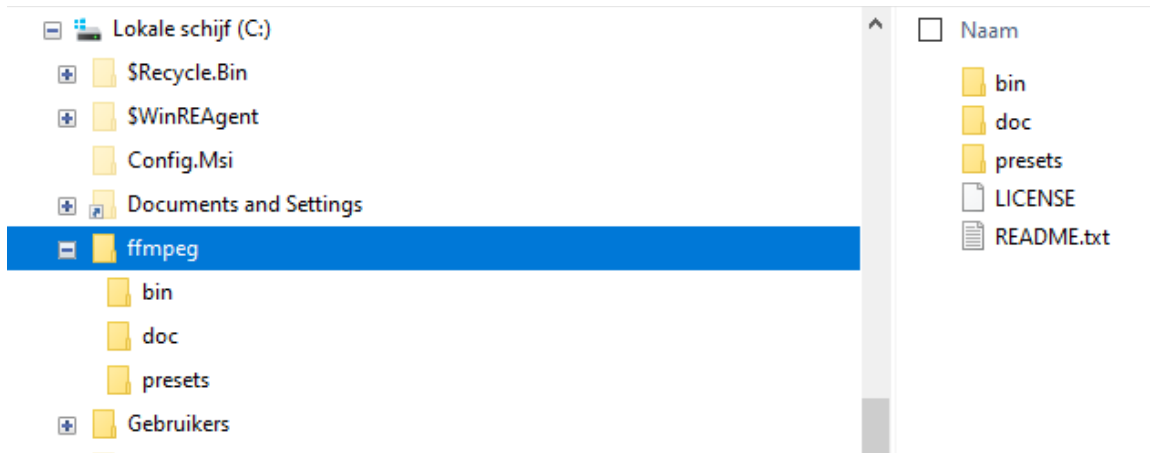
How to use Whisper in Python

Mar 26, 2024 AI

Whisper is an AI model from OpenAI that allows you to convert any audio to text with high quality and accuracy. In this article I will show you how to use this AI model to get transcriptions from an audio file and how to run it with Python.

FFmpeg is needed

To use the Whisper API with Python, you'll need to install [FFmpeg](#) on your laptop. [This WikiHow guide](#) provides clear, step-by-step instructions for setup. I followed it on a laptop running Windows 10 Pro, and here's what the setup looked like once completed.



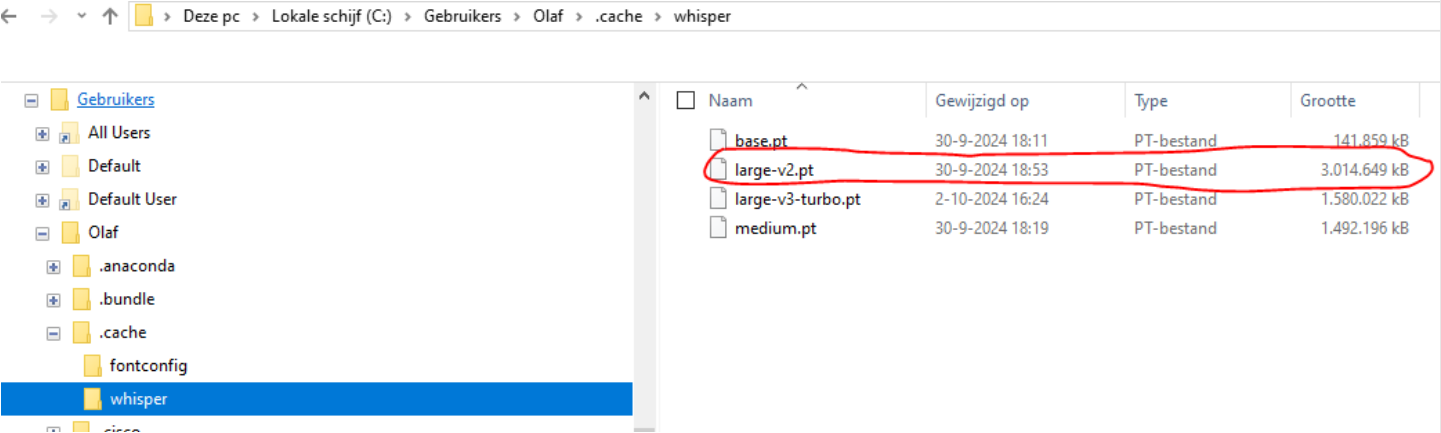
Offline use, so privacy friendly

When you run this piece of Python code for the first time,

```
# main.py
import whisper
model = whisper.load_model('large')
```

the 'large' model is downloaded to your machine once. (See here for [the available models](#).) To my great surprise, this turned out to be just a single 3GB file, handling all speech-to-text tasks, without needing any further internet connection. So no smoking data centers, overheated servers, or massive cloud computing power, but just a file on your own computer that you can use offline. Best of all, it's great for privacy, as all processing happens entirely on your own device, ensuring your data stays private and secure.

Here's a screenshot of the model on my home laptop. What happens inside that .pt file is pure magic!



Speed

Does transcription run at a reasonable speed? With the 'large-v2' model I'm using, transcription operates at roughly real-time, so a 15-minute audio file takes about 15-20 minutes to process. Smaller models, like 'base' and 'medium,' are faster but typically produce lower-quality transcriptions.

Great quality! With subtitles! Even with poor input!

Besides Whisper's offline capabilities, I am utterly amazed by the quality of the generated text. I can show this best through this (rather dull and quite lengthy) [test video](#) in which I used myself as the test subject:

The unformatted text block in the file description was generated entirely by Whisper, with only minimal human post-corrections. Take note of how accurately it handles named entities, technical terms, and proper capitalization, truly impressive!

called Wikimedia Commons. And the volunteers of creators, or sorry, the communities of creators, mostly volunteers, that work on the benefit. For instance, we as the KB take a lot of information from Wikipedia, Wikidata, but also make sure that we contribute to Wik heritage institutions and other universities about Wikidata. This is exactly the thing that I'm doing right now. So I'm sharing my knowl do I hope that you will have learned or gained some more understanding of by the end of this afternoon? First of all, that you have a b also from the community/social perspective. So that you will also have a little insight in how the community around Wikidata is organ this real hard data aspect of Wikidata. But for me, the community aspects are equally important, actually. And actually, I think they'r presentation. Then next, the second learning goal is trying to give you a small insight into how Wikidata can be relevant for research, which a separate video is available also, is trying to guide you through making your first steps into Wikidata yourself. So doing your ourselves, what is Wikidata? Well, if you talk with people about Wikidata, you get a wide variety of answers. But depending on their Wikidata is that it is a database containing structured descriptions of all sorts of things. And what kinds of things are, can you think hit single by Girlfriend, or this book, or this American judge, or this airport in the United States, or this gene, or this election in Catal the Netherlands, or this family name, or this quasar in the constellation of Ursa Major, or this really nice car, or this really nice metal randomly chosen library from the Netherlands. So you see there are a lot of different things described in Wikidata. In total, when I las

In the video, you can tell I wasn't making an effort to speak clearly, loudly, enthusiastically, or fluently. Yet, despite these less-than-ideal inputs, Whisper still managed to produce a fantastic transcription using just that 3GB .pt file (and FFmpeg). Absolutely amazing!

And the [subtitles \(closed captions\)](#) you see in the video were also completely generated by Whisper, in which all timings are spot-on as well.

Example code, try it yourself

To share my knowledge and code, I created the GitHub repo <https://github.com/KBNLresearch/videotools>

The relevant module is [transcribe_audio.py](#), which is run from [runtools.py](#), the main function of this repo.

If you want, you can have the audio transcript corrected by ChatGPT, for which I made an initial setup in [ai_correct_audiotranscripts.py](#). To use this, you'll need an [OpenAI API key](#). But please note that you'll lose the privacy advantage and offline use, as the ChatGPT models are far too large to run on a personal laptop.

As a side product, I also created a few other video and audio tools that only require FFmpeg, without a need for Whisper or ChatGPT.

Feedback is welcome!

Since this was just a first experiment with this new piece of AI for me, I'd love to hear your questions, feedback, tips, etc. You can find my contact details below.

Similar articles

- [Super efficient! Subtitling or transcribing your video with AI](#) (in Dutch)
- [This article on Github](#) and [Zenodo](#), 6 November 2024

A collection of video and audio processing tools



Description

This repo performs various operations on video and audio files, including:

1. Extracting short video clips from longer ones.
2. Enhancing audio by adjusting pitch and volume, eg. for a deeper voice.
3. Compressing and converting video files to WebM format.
4. Extracting audio from a video and saving it as an MP3 file.
5. Amplifying audio if necessary.
6. Transcribing audio using Whisper.
7. Correcting raw audio transcripts using ChatGPT.
8. Embedding subtitles into the WebM video files.

Contact

The [Videotools repo](#) is developed and maintained by Olaf Janssen, Wikimedia coordinator [@KB, national library of the Netherlands](#). You can find his contact details on his [KB expert page](#) or via his [Wikimedia user page](#).

KB } national library
of the netherlands

Licensing

All original materials in this repo, except for the [blog article header](#), are released under the [CC0 1.0 Universal license](#), effectively donating all original content to the public domain.

