



Introduction to

SPEECH RECOGNITION



with

 OpenAI
Whisper 

ASR

.mp3 to TXT

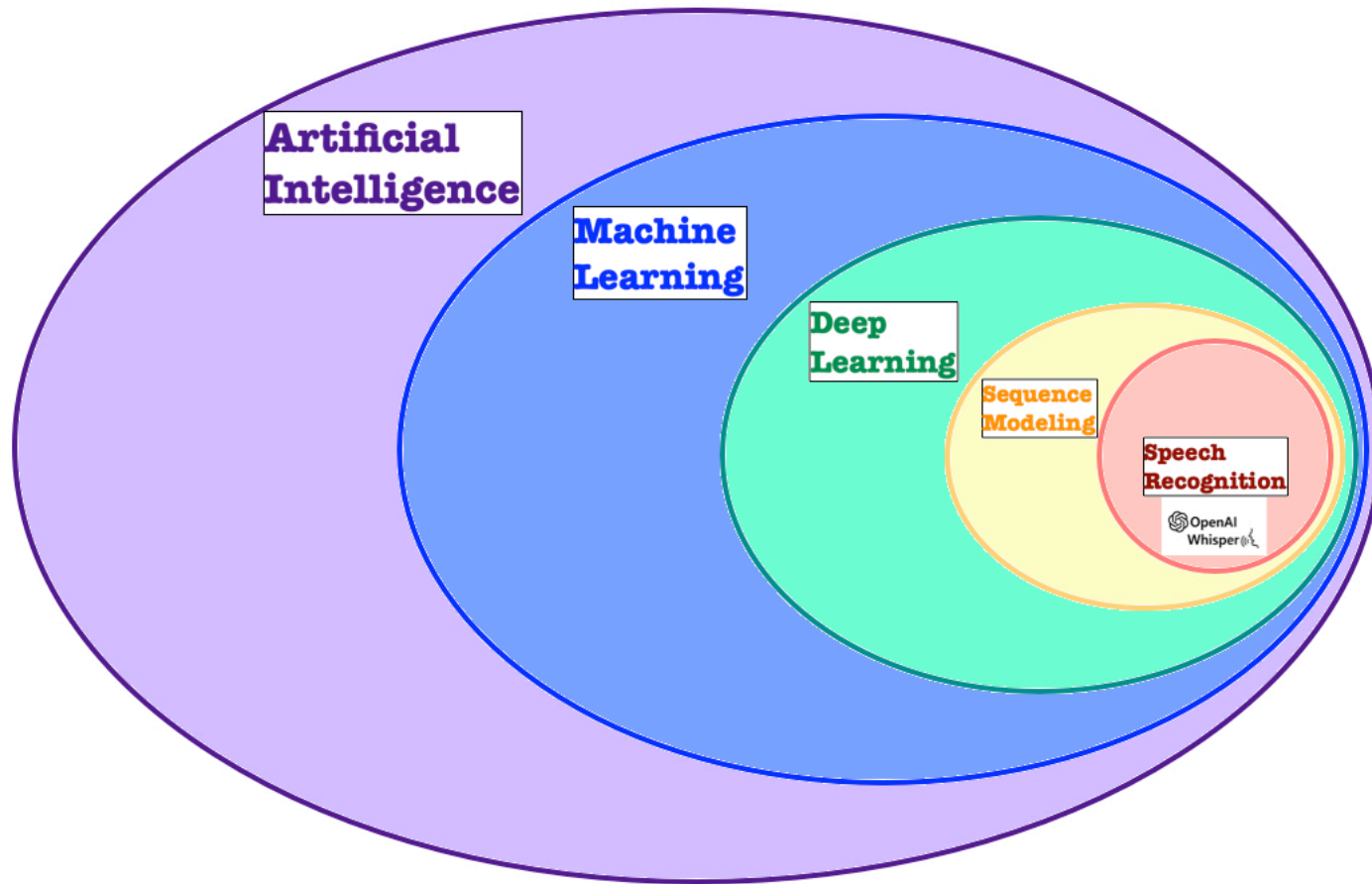
.wav to TXT

 LIVE

Song LYRICS maker!
No Movie Subtitles No Problem!



Category of Speech Recognition



AI → Machine Learning → Deep Learning → Sequence Modeling → Speech Recognition

Artificial Intelligence

The broad science of creating systems that can perform tasks typically requiring human intelligence.

Machine Learning

A subset of AI where machines learn from data to improve performance on a task without being explicitly programmed.

Deep Learning

A subset of Machine Learning based on neural networks with many layers (hence “deep”).

Sequence Modeling

A class of deep learning focused on data that comes in sequences — where order matters.

Speech Recognition

The application of sequence modeling—a branch of deep learning—to convert speech into text.

Workflow:

1. Audio input (e.g., microphone)
2. Preprocessing
3. Sequence model analyzes the waveform over time
4. Decodes speech → words/text

From Audio to Text: Let's Build Automatic Speech Recognition (ASR) now!

Let's generate the Lyrics for "Fight Song" by Rachel Platten

Environment Setting

Step 1: Create a Virtual Environment

```
python -m venv whisper-env
# Then activate it:

# macOS / Linux
source whisper-env/bin/activate

# Windows
whisper-env\Scripts\activate
```

Step 2: Install Required Libraries: Torch, Whisper, and FFmpeg

```
pip install torch

pip install faster-whisper

#For macOS Users:    brew install ffmpeg

#For Linux Users:    sudo apt install ffmpeg

#For Windows Users:  Use installer from ffmpeg.org and add to PATH
```

With OpenAI Whisper we get a harmless warning when using CPU:

UserWarning: FP16 is not supported on CPU; using FP32 instead
warnings.warn("FP16 is not supported on CPU; using FP32 instead")

If you don't hate warnings, you can just disregard it or suppress it.

But if you're allergic about warnings, you can use Faster-Whisper:

Use `compute_type="float32"` or `"int8"` in Faster-Whisper to avoid CPU warnings.


```

1  from faster_whisper import WhisperModel # type: ignore
2
3  # =====
4  # 1. INPUT
5  # =====
6  # Specify the input audio file
7  input_audiofile = "sample1.mp3"
8
9  # =====
10 # 2. LOAD MODEL (Sequence Modeling Engine)
11 # =====
12 # Load the Faster-Whisper model with optimized compute_type for CPU (no FP16 warning)
13 # compute_type options: "int8", "int8_float16", "float16", "float32"
14 model = WhisperModel("base", compute_type="float32")
15
16 # =====
17 # 3. FULL PIPELINE: Preprocessing + Sequence Modeling + Decoding
18 # =====
19 # Transcribe the audio file:
20 # - Preprocessing (audio loading, resampling) is handled automatically
21 # - Sequence modeling is done using optimized CPU inference
22 # - Decoding converts audio into readable text
23 print(f"Transcribing '{input_audiofile}'...")
24 segments, info = model.transcribe(input_audiofile, language="en")
25
26 # Combine all segments into a single string
27 lyrics = ""
28 for segment in segments:
29     lyrics += segment.text.strip() + "\n"
30
31 # =====
32 # 4. OUTPUT
33 # =====
34 # Specify the output text filename
35 output_textfile = input_audiofile.replace(".mp3", ".txt")
36 # Save the transcribed text to a .txt file
37 with open(output_textfile, "w", encoding="utf-8") as f:
38     f.write(lyrics)
39
40 print(f"Transcription complete. Lyrics saved to '{output_textfile}'")

```

What happens when you run the program for the first time?

Faster-Whisper model components are downloaded from the cloud.

config.json describes the model's architecture (e.g., number of layers, hidden size, attention heads)

vocabulary.txt contains a list of tokens/words the model recognizes – essential for decoding

tokenizer.json contains rules for converting raw audio/text \leftrightarrow token IDs.

It handles both directions — it's the bridge between text and the model's numerical world.

```
gabriel@MacBookAir-2 speech-recognition % python 01_audio_to_text_file.py
config.json: 100%|██████████████████████████████████████████████████████| 2.31k/2.31k [00:00<00:00, 12.3MB/s]
vocabulary.txt: 100%|██████████████████████████████████████████████████████| 460k/460k [00:13<00:00, 35.3kB/s]
tokenizer.json: 100%|██████████████████████████████████████████████████████| 2.20M/2.20M [01:17<00:00, 28.4kB/s]
tokenizer.json: 100%|██████████████████████████████████████████████████████| 2.20M/2.20M [01:17<00:00, 28.4kB/s]
model.bin: 100%|██████████████████████████████████████████████████████| 145M/145M [01:55<00:00, 621kB/s]
Transcribing 'sample1.mp3'...
Transcription complete. Lyrics saved to 'sample1.txt'
```

Two Key Roles of tokenizer.json:

1) Converts text (e.g., "muraho neza") into a list of token IDs the model understands (e.g., [4852, 2091, 1245])

Why? Neural networks can't understand words — only numbers. So we “tokenize” the text into IDs.

2) Converts model's predicted token IDs (e.g., [4852, 2091, 1245]) back into readable text ("muraho neza")

Why? After the model "thinks" in token IDs, we need to turn those IDs back into something humans can read.

To explore more, I changed the directory to the cache (check the path to the cache on your PC):





```
cd /Users/gabriel/.cache/huggingface/hub/models--Systran--faster-whisper-base/snapshots/ebe41f70d5b6dfa9166e2c581c45c9c0cfc57b66
```

Hugging Face

Hugging Face is the platform powering modern AI models, making it easy to use, train, and share speech recognition systems — including Whisper.

It's "Model Hub" can be thought as a GitHub for AI models

We can:

-  Search for existing models (Whisper, Wav2Vec2, XLS-R, etc.)
-  Download pretrained weights (what our script does the first time)
-  Upload your fine-tuned models so our students (or the world) can use them!
-  Example: <https://huggingface.co/openai/whisper-large>

For Code and Documentation

Visit <https://github.com/benax-rw/asr>