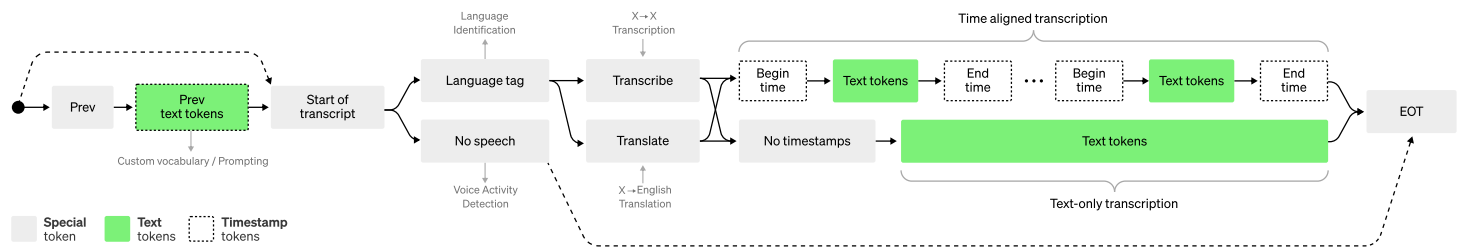


Speech-to-Text with Whisper AI



Housekeeping

1. Check that the recording is on
2. Check audio and screenshare
3. Share link to notebook in chat
4. Light mode and readable font size
5. GPU runtime and `run-all`

Some Terminology

- **Speech-To-Text (STT):** A task for taking an audio file with speech as input, and returning the words and sentences spoken as the output, usually with timestamps.
- **Transcripts:** A file with all the audio saved in a text format.
- **(Close) Captions:** Text that follows the audio, and may include descriptions of the audio and video content.
- **Subtitles:** translations of captions into another language.
- **Word Error Rate (WER):** A metric used to evaluate transcription quality. It is the percentage of words incorrectly transcribed in utterances in a transcript, per 100 words in the transcript.

✓ Transcription formats and content

- .vtt, .srt
- Textfiles, Json, textgrids
- Speaker, content, timestamps

1. VTT (WebVTT) WebVTT is commonly used for displaying timed text tracks in HTML5 videos.

WEBVTT

```
00:00:00.000 --> 00:00:02.500
Hello, and welcome to today's workshop

00:00:02.500 --> 00:00:05.000
where we will discuss speech recognition.
```

2. SRT (SubRip Subtitle) SRT is one of the most widely used subtitle formats, known for its simplicity.

SRT

```
1
00:00:00,000 --> 00:00:02,500
Hello, and welcome to today's workshop

2
00:00:02,500 --> 00:00:05,000
where we will discuss speech recognition.
```

3. JSON can be useful for storing structured data, including transcription with timestamps.

```
{
  "transcriptions": [
    {
      "start": "00:00:00.000",
```

```

        "end": "00:00:02.500",
        "text": "Hello, and welcome to our video."
    },
    {
        "start": "00:00:02.500",
        "end": "00:00:05.000",
        "text": "Today, we will discuss the basics of speech recognition."
    }
]
}

```

4. Textgrids- these are outputs from speech processing software like Praat, where the same file contains data from multiple annotation tiers. Its a great way to annotate audio more than one way.

```

FileType = "ooTextFile"
ObjectClass = "TextGrid"
xmin = 0
xmax = 10
tiers? <exists>
size = 2
item []:
    item [0]:
        class = "IntervalTier"
        name = "Words"
        xmin = 0
        xmax = 10
        intervals: size = 3
        intervals [0]: xmin = 0.0; xmax = 2.5; text = "Hello"
        intervals [1]: xmin = 2.5; xmax = 5.0; text = "and welcome"
        intervals [2]: xmin = 5.0; xmax = 10.0; text = "to today's session"
    item [1]:
        class = "IntervalTier"
        name = "Phonemes"
        xmin = 0
        xmax = 10
        intervals: size = 6
        intervals [0]: xmin = 0.0; xmax = 0.5; text = "H"

```

```
intervals [1]: xmin = 0.5; xmax = 1.0; text = "ε"  
intervals [2]: xmin = 1.0; xmax = 1.5; text = "l"  
intervals [3]: xmin = 1.5; xmax = 2.0; text = "o"  
intervals [4]: xmin = 2.5; xmax = 3.0; text = "a"  
intervals [5]: xmin = 3.0; xmax = 3.5; text = "n"
```

Bottom line: all transcription outputs contain information about the content of a recording, and can be inter-converted. For our NLP pipeline, it is important to know which one is being asked for.

Popular use cases

- Accessibility
- Audio input for assistants, hands-free applications
- Downstream NLP tasks

Available Transcription services

- Free:
 - Coqui
 - SpeechBrain
- Paid
 - Zoom (free for U of A affiliates)
 - Amazon Web Services (AWS)
 - Google ASR

Zoom Transcription and Captions

- Free for premium accounts
- Setup on the Zoom Cloud
- Great for non-private settings
- Generates `.vtt` files with timestamps, as well as transcript file `.txt`. Ignores false starts and filler sounds
- Caption support for many languages

Documentation:

- [Enabling or disabling audio transcription for cloud recordings](#)
- [Enabling automated captions](#)
- [Enabling and configuring translated captions](#)
- [Real-time automatic caption translation](#)

Downstream NLP tasks from STT data

- Conversation summarization and automatic note-taking
- Topic analysis
- Named Entity Recognition (NER)
- Speaker dominance and conversation quality assessments

✓ Transformers and their impact on ASR

Speech and Language models

- Language Models:
 - Language models work with written or transcribed text.
 - They predict the likelihood of a sequence of words.
 - They generate coherent text based on input prompts.
 - They are trained on large text corpora to understand grammar, context, and semantics.
 - Applications: text completion, machine translation, conversation generation.
- Speech Models:
 - Focus: processing and comprehending spoken language.
 - They convert audio signals (mel-frequency spectrograms, MFCCs) into text.
 - They can also process audio for other tasks (language/dialect detection, voice activity detection, speaker identification and diarization, audio source detection).
 - They can identify linguistic units like phonemes, words, utterances, as well as non-speech sounds (background noise, animal sounds).
 - They can process other nuances of spoken human interactions, such as tone and prosody, accent, and pronunciation.

Current SOTA models in both domains use the transformer architecture, and work with an objective to predict the next sequence for a given unit of data. While speech models are primarily concerned with audio input and the acoustic features of spoken language, language models focus on the syntactic and semantic structures of written language.

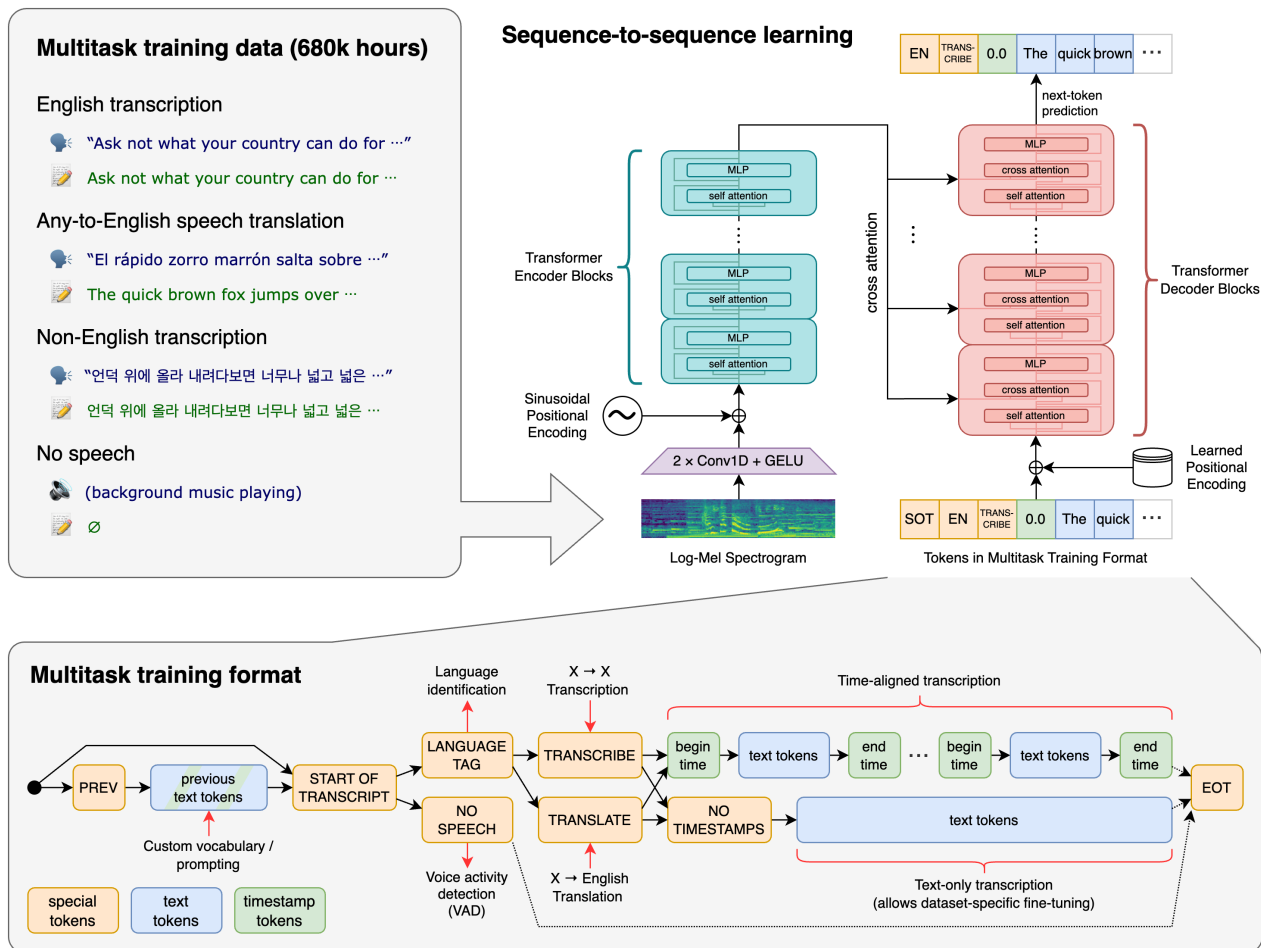
Note: Human speech almost always has the aim of linguistic communication. On the surface, speech models may be only processing the audio signal. But they are often trained and applied for using the linguistic units present in the spoken data. In most cases, speech models are not independent of language models.

Wav2Vec 2.0 (Facebook AI, 2020)

- Self-supervised learning model developed by Facebook AI Research (FAIR) for automatic speech recognition (ASR).
- Pre-trained on vast amounts of unlabeled audio data- learnt rich representations of speech signals.
- Two-step process:
 - First, it learns to predict masked portions of the audio waveform from the unmasked segments (pre-training).
 - Next, the model is fine-tuning on smaller labeled datasets to improve its performance on specific ASR tasks.
- Thus, it leverages both self-supervised learning and supervised learning effectively.
- It captures patterns in speech without a lot of labeled data.

[Paper](#)

✓ Web-scale Supervised Pretraining for Speech Recognition (Whisper)



[image source](#)

- Powerful audio transformer model from OpenAI.
- This model maps utterances and their transcribed form across multiple languages.
- It can be downloaded and used on one's own setup (GPU needed) without sending data through the web.
- Its training data includes many different recording conditions: noisy and quiet environments, audio with and without speech, songs, etc.
- So it performs well on both quiet and noisy environments.
- Whisper used a **sequence-to-sequence transformer** model.
- It also uses weak supervision for training on transcripts (that is, not all of the transcripts are labelled or even generated by humans).
- Its speech model uses a 'multitask training format' and a set of special tokens that can understand the audio data collectively for a lot of tasks.
- It is powerful because the model has been pre-trained on many speech processing tasks,

such as multilingual speech recognition, speech translation, spoken language identification, and voice activity detection.

When we call the model to process a file, it makes predictions for the set of tasks as a whole, instead of sending the data through different stages.

Data and pipeline

- 680,000 hours of audio and transcripts
- Source of data: the internet.
- 65% (438,000 hours) English-language audio
- ~ 18% (126,000 hours) non-English audio, English translations
- ~ 17% (117,000 hours) non-English audio and transcripts from 98 languages.

Input audio is split into 30-second chunks, converted into a log-Mel spectrogram, and then passed into an encoder. A decoder is trained to predict the corresponding text caption, intermixed with special tokens that direct the single model to perform tasks such as language identification, phrase-level timestamps, multilingual speech transcription, and to-English speech translation.

Setup

We use Whisper by calling the python library, and downloading the necessary language model.

```
! pip install git+https://github.com/openai/whisper.git -q

# Load the model
import whisper
model = whisper.load_model("base")
```

✓ STT with Whisper

In this example, we will run Whisper on the command line for a few .mp3 files in English and Korean. Based on the size of the file, the model may need more or less time.

```
import locale
print(locale.getpreferredencoding())
import locale
def getpreferredencoding(do_setlocale = True):
    return "UTF-8"
locale.getpreferredencoding = getpreferredencoding
```

⇄ utf-8

```
# Get some audio files
!wget -O mary.mp3 https://raw.githubusercontent.com/petewarden/openai-whisper-web/
!wget -O daisy_HAL_9000.mp3 https://raw.githubusercontent.com/petewarden/openai-w
!wget -O AllStar.mp3 https://raw.githubusercontent.com/keatonkraiger/Whisper-Tran
!wget -O Cupid_Fifty_Fifty_Korean_Version.mp3 https://raw.githubusercontent.com/k
```

→ --2025-04-02 20:29:23-- <https://raw.githubusercontent.com/petewarden/openai-voice-to-voice-1/raw/main/20250402202923mary.mp3>
 Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133
 Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133:443
 HTTP request sent, awaiting response... 200 OK
 Length: 100483 (98K) [audio/mpeg]
 Saving to: 'mary.mp3'

mary.mp3 100%[=====>] 98.13K --.-KB/s in 0.003s

2025-04-02 20:29:23 (35.5 MB/s) - 'mary.mp3' saved [100483/100483]

--2025-04-02 20:29:23-- <https://raw.githubusercontent.com/petewarden/openai-voice-to-voice-1/raw/main/20250402202923daisy.mp3>
 Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133
 Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133:443
 HTTP request sent, awaiting response... 200 OK
 Length: 150436 (147K) [audio/mpeg]
 Saving to: 'daisy_HAL_9000.mp3'

daisy_HAL_9000.mp3 100%[=====>] 146.91K --.-KB/s in 0.003s

2025-04-02 20:29:24 (45.3 MB/s) - 'daisy_HAL_9000.mp3' saved [150436/150436]

--2025-04-02 20:29:24-- <https://raw.githubusercontent.com/keatonkraiger/Whisper-to-Whisper-1/raw/main/20250402202924AllStar.mp3>
 Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133
 Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133:443
 HTTP request sent, awaiting response... 200 OK
 Length: 4823469 (4.6M) [audio/mpeg]
 Saving to: 'AllStar.mp3'

AllStar.mp3 100%[=====>] 4.60M --.-KB/s in 0.03s

2025-04-02 20:29:25 (133 MB/s) - 'AllStar.mp3' saved [4823469/4823469]

--2025-04-02 20:29:25-- https://raw.githubusercontent.com/keatonkraiger/Whisper-to-Whisper-1/raw/main/20250402202925Cupid_Fifty_Fifty_Korean_Version.mp3
 Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 185.199.108.133
 Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|185.199.108.133:443
 HTTP request sent, awaiting response... 200 OK
 Length: 4867062 (4.6M) [audio/mpeg]
 Saving to: 'Cupid_Fifty_Fifty_Korean_Version.mp3'

Cupid_Fifty_Fifty_K 100%[=====>] 4.64M --.-KB/s in 0.03s

2025-04-02 20:29:25 (141 MB/s) - 'Cupid_Fifty_Fifty_Korean_Version.mp3' saved [4867062/4867062]

```
from IPython.display import Audio
Audio("/content/mary.mp3")
```



0:00

-0:08

```
from IPython.display import Audio
Audio("/content/daisy_HAL_9000.mp3")
```



0:00

0:48

```
from IPython.display import Audio
Audio("/content/AllStar.mp3")
```



0:00

-3:20

```
from IPython.display import Audio
Audio("/content/Cupid_Fifty_Fifty_Korean_Version.mp3")
```



0:00

-3:22

✓ Generating transcription files

[Whisper documentation](#)

[Code Source](#)

```
# install whisper from the Github repository:
!pip install git+https://github.com/openai/whisper.git -q
```

```

🔄 Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing metadata (pyproject.toml) ... done
_____ 1.2/1.2 MB 41.1 MB/s eta 0:00:00
_____ 363.4/363.4 MB 4.3 MB/s eta 0:00:00
_____ 13.8/13.8 MB 65.4 MB/s eta 0:00:00
_____ 24.6/24.6 MB 35.4 MB/s eta 0:00:00
_____ 883.7/883.7 kB 41.6 MB/s eta 0:00:00
_____ 664.8/664.8 MB 3.3 MB/s eta 0:00:00
_____ 211.5/211.5 MB 5.3 MB/s eta 0:00:00
_____ 56.3/56.3 MB 13.0 MB/s eta 0:00:00
_____ 127.9/127.9 MB 7.4 MB/s eta 0:00:00
_____ 207.5/207.5 MB 5.5 MB/s eta 0:00:00
_____ 21.1/21.1 MB 63.8 MB/s eta 0:00:00
Building wheel for openai-whisper (pyproject.toml) ... done

```

```
# Other tools for processing audio files:
```

```
!apt install ffmpeg
!pip install setuptools-rust
```

```

🔄 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 30 not upgraded.
Collecting setuptools-rust
  Downloading setuptools_rust-1.11.0-py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: setuptools>=62.4 in /usr/local/lib/python3.11/
Collecting semantic_version<3,>=2.8.2 (from setuptools-rust)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl.metadata (9.7 kB)
  Downloading setuptools_rust-1.11.0-py3-none-any.whl (27 kB)
  Downloading semantic_version-2.10.0-py2.py3-none-any.whl (15 kB)
Installing collected packages: semantic_version, setuptools-rust
Successfully installed semantic_version-2.10.0 setuptools-rust-1.11.0

```

```
# Load the model
```

```
import whisper
```

```
model = whisper.load_model("base")
```

```

🔄 100%|████████████████████████████████████████████████████████████████████████████████| 139M/139M [00:01<00:00, 77.4MiB,

```

Check usage guide for more:

!whisper --help



<https://arxiv.org/abs/1609.08144>, uses simple length r
default (default: None)

--suppress_tokens SUPPRESS_TOKENS
comma-separated list of token ids to suppress during ;
suppress most special characters except common punctua

--initial_prompt INITIAL_PROMPT
optional text to provide as a prompt for the first wir

--carry_initial_prompt CARRY_INITIAL_PROMPT
if True, prepend initial_prompt to every internal dec
reduce the effectiveness of condition_on_previous_text

--condition_on_previous_text CONDITION_ON_PREVIOUS_TEXT
if True, provide the previous output of the model as a
window; disabling may make the text inconsistent across
model becomes less prone to getting stuck in a failure
True)

--fp16 FP16
whether to perform inference in fp16; True by default

--temperature_increment_on_fallback TEMPERATURE_INCREMENT_ON_FALLBACK
temperature to increase when falling back when the dec
either of the thresholds below (default: 0.2)

--compression_ratio_threshold COMPRESSION_RATIO_THRESHOLD
if the gzip compression ratio is higher than this valu
decoding as failed (default: 2.4)

--logprob_threshold LOGPROB_THRESHOLD
if the average log probability is lower than this valu
decoding as failed (default: -1.0)

--no_speech_threshold NO_SPEECH_THRESHOLD
if the probability of the <|nospeech|> token is higher
the decoding has failed due to `logprob_threshold`, co
as silence (default: 0.6)

--word_timestamps WORD_TIMESTAMP
(experimental) extract word-level timestamps and refin
on them (default: False)

--prepend_punctuations PREPEND_PUNCTUATIONS
if word_timestamps is True, merge these punctuation sy
word (default: "'\"?,:;!@<>[]{}~)

--append_punctuations APPEND_PUNCTUATIONS
if word_timestamps is True, merge these punctuation sy
previous word (default: "'\"?,:;!@<>[]{}~)

--highlight_words HIGHLIGHT_WORDS
(requires --word_timestamps True) underline each word
srt and vtt (default: False)

--max_line_width MAX_LINE_WIDTH
(requires --word_timestamps True) the maximum number o
line before breaking the line (default: None)

--max_line_count MAX_LINE_COUNT
(requires --word_timestamps True) the maximum number o
(default: None)

```

--max_words_per_line MAX_WORDS_PER_LINE
    (requires --word_timestamps True, no effect with --max_speakers)
    maximum number of words in a segment (default: None)
--threads THREADS
    number of threads used by torch for CPU inference; supported on
    MKL_NUM_THREADS/OMP_NUM_THREADS (default: 0)
--clip_timestamps CLIP_TIMESTAMPS
    comma-separated list start,end,start,end,... timestamp pairs
    clips to process, where the last end timestamp defaults to the end of the
    file (default: 0)
--hallucination_silence_threshold HALLUCINATION_SILENCE_THRESHOLD
    (requires --word_timestamps True) skip silent periods longer than the
    threshold (in seconds) when a possible hallucination is detected
    (default: 0)

```

```

# Check for GPU availability:
!nvidia-smi

```

🔗 Wed Apr 2 20:31:49 2025

| | | | | | | | |
|----------------------|----------|------|---------------|---------------------------|--------------|----------------------|--|
| NVIDIA-SMI 550.54.15 | | | | Driver Version: 550.54.15 | | CUDA Version: 12.4.1 | |
| GPU | Name | | Persistence-M | Bus-Id | Disp.A | Volatile | |
| Fan | Temp | Perf | Pwr:Usage/Cap | | Memory-Usage | GPU-Util | |
| ===== | | | | | | | |
| 0 | Tesla T4 | | Off | 00000000:00:04.0 | Off | | |
| N/A | 54C | P0 | 30W / 70W | 544MiB / 15360MiB | | 0% | |
| ===== | | | | | | | |

| Processes: | | | | | | |
|------------|----|----|-----|------|--------------|--|
| GPU | GI | CI | PID | Type | Process name | |
| | ID | ID | | | | |
| ===== | | | | | | |

Process an audio file with specified parameters:

```
!whisper /content/mary.mp3 \
--model medium --task transcribe \
--output_dir mary_transcription \
--output_format all \
--word_timestamps True
```

```
⇒ 100%|████████████████████████████████████████| 1.42G/1.42G [00:20<00:00, 73.5MiB,
Detecting language using up to the first 30 seconds. Use `--language` to spec:
Detected language: English
[00:01.140 --> 00:08.240] Mary had a little lamb, its fleece was white as sn
```

Process an audio file with specified parameters:

```
!whisper /content/AllStar.mp3 \
--model medium \
--task transcribe \
--output_dir AllStar_transcription \
--output_format all \
--word_timestamps True
```

You can also try:

```
# --output_format srt
# --max_words_per_line 3
```

```
⇒ [00:14.580 --> 00:18.440] And the shape of an L on her forehead
[00:19.240 --> 00:22.120] Well, the years start coming and they don't stop c
[00:22.120 --> 00:24.460] Fed to the rules and I hit the ground running
[00:24.460 --> 00:26.620] Didn't make sense not to live for fun
[00:26.620 --> 00:28.960] Your brain gets smart but your head gets dumb
[00:28.960 --> 00:31.700] So much to do, so much to see, so what's wrong?
[00:32.020 --> 00:33.400] We're taking the back streets
[00:33.400 --> 00:35.340] You'll never know if you don't go
[00:35.960 --> 00:37.960] You'll never shine if you don't glow
[00:37.960 --> 00:40.360] Hey now, you're an all-star
[00:40.360 --> 00:42.800] Get your game on, go play
[00:42.800 --> 00:44.940] Hey now, you're a rock star
[00:44.940 --> 00:47.400] Get the show on, get paid
[00:47.400 --> 00:50.820] And all that's left in us is gold
[00:50.820 --> 00:55.940] Only shooting stars break the mold
[00:56.440 --> 00:58.980] It's a cool place and they say it gets colder
[00:58.980 --> 01:01.240] You're bundled up now, wait till you get older
[01:01.240 --> 01:03.160] But the media men beg to differ
[01:03.160 --> 01:05.340] Judging by the hole in the satellite picture
[01:05.340 --> 01:07.580] The ice we skate is getting pretty thin
```



```

[01:07.580 --> 01:09.980] The water's getting warm so you might as well swim
[01:09.980 --> 01:12.220] But worlds on fire, how about yours?
[01:12.620 --> 01:14.820] That's the way I like it and I'll never get bored
[01:14.820 --> 01:17.240] Hey now, you're an all-star
[01:17.240 --> 01:19.700] Get your game on, go play
[01:19.700 --> 01:21.880] Hey now, you're a rock star
[01:21.880 --> 01:24.360] Get the show on, get paid
[01:24.360 --> 01:27.840] And all that's left in us is gold
[01:27.840 --> 01:32.360] Only shooting stars break the mold
[01:45.960 --> 01:47.700] Go for the moon set
[01:52.360 --> 01:54.180] Hey now, you're an all-star
[01:54.180 --> 01:56.680] Get your game on, go play
[01:56.680 --> 01:58.720] Hey now, you're a rock star
[01:58.720 --> 02:01.240] Get the show on, get paid
[02:01.240 --> 02:04.880] And all that's left in us is gold
[02:04.880 --> 02:07.240] Only shooting stars
[02:08.240 --> 02:11.680] Somebody once asked could I spare some change
[02:11.680 --> 02:16.120] Well guess I need to get myself away from this place
[02:16.120 --> 02:19.200] I said yeah, what a concept
[02:19.200 --> 02:21.500] I could use a little fuel myself
[02:21.500 --> 02:26.220] And we could all use a little change
[02:26.220 --> 02:29.040] Well, the years start coming and they don't stop coming
[02:29.040 --> 02:31.380] Fed to the rules that I'll hit the ground running
[02:31.380 --> 02:33.580] Didn't make sense not to live for fun
[02:33.580 --> 02:35.960] Your brain gets smart but your head gets dumb
[02:35.960 --> 02:38.620] So much to do, so much to see, so what's wrong?
[02:38.780 --> 02:40.480] We're taking the backstreets
[02:40.480 --> 02:42.500] You'll never know if you don't go
[02:42.500 --> 02:44.820] You'll never shine if you don't glow
[02:44.820 --> 02:47.240] Hey now, you're an all-star
[02:47.240 --> 02:49.680] Get your game on, go play
[02:49.680 --> 02:51.800] Hey now, you're a rock star
[02:51.800 --> 02:54.340] Get the show on, get paid
[02:54.340 --> 02:57.760] And all that's left in us is gold
[02:57.760 --> 02:59.760] Only shooting stars
[03:00.340 --> 03:02.740] Break the mold
[03:02.740 --> 03:06.880] And all that's left in us is gold
[03:06.880 --> 03:08.760] Only shooting stars
[03:09.580 --> 03:12.040] Break the mold

```

✓ Speech translation

[Code Source](#)

```
# Process and translate audio from Korean, and save an English transcript:
!whisper /content/Cupid_Fifty_Fifty_Korean_Version.mp3 \
--language Korean \
--task translate \
--model medium \
--output_dir Cupid_Fifty_Fifty_Korean_translation \
--output_format srt
```

```
➡ [00:30.000 --> 00:35.000] Tell me honey, I'm crying in my room again
[00:35.000 --> 00:41.000] I want to hide it, but still I want it more, more,
[00:41.000 --> 00:45.000] I give a second chance, I'm cute but
[00:45.000 --> 00:50.000] I believe in myself, I'm really stupid
[00:50.000 --> 00:55.000] I'll show you, hide it, the love is real
[00:55.000 --> 00:59.000] I feel it, it is so dumb
[00:59.000 --> 01:03.000] I walk my dream again, everyday
[01:03.000 --> 01:07.000] When I wake up, it's more fluid
[01:07.000 --> 01:09.000] Waiting around is always
[01:09.000 --> 01:11.000] Honestly, I'm comfortable now
[01:11.000 --> 01:14.000] Is it as thrilling as I imagined?
[01:14.000 --> 01:16.000] Now I'm so lonely
[01:16.000 --> 01:20.000] I practiced in my dreams every day, kiss me
[01:20.000 --> 01:25.000] Actually, I'm crying in my room again
[01:25.000 --> 01:29.000] I want to hide it, but still I want it more, more,
[01:29.000 --> 01:33.000] I give a second chance, I'm cute but
[01:33.000 --> 01:38.000] I believe in myself, I'm really stupid
[01:38.000 --> 01:42.000] I'll show you, hide it, the love is real
[01:42.000 --> 01:47.000] I feel it, it is so dumb
[01:47.000 --> 01:49.000] I'm so lonely, hold me tight
[01:49.000 --> 01:51.000] I want something thrilling, really love me truly
[01:51.000 --> 01:53.000] There's no need to wait for me, I can't wait
[01:53.000 --> 01:55.000] I don't believe anymore, now I'm gonna make it mine
[01:55.000 --> 01:57.000] Love is a lie, our human life is right
[01:57.000 --> 01:59.000] It's not a joke, so give it to me right now
[01:59.000 --> 02:01.000] No more chance to you
[02:01.000 --> 02:03.000] You know, hey da da da dumb boy
[02:03.000 --> 02:07.000] In my dreams every night
[02:07.000 --> 02:11.000] Someone who will share this feeling
[02:11.000 --> 02:19.000] I'm a fool, a fool for love, a fool for love
[02:19.000 --> 02:23.000] I give a second chance, I'm cute but
[02:23.000 --> 02:28.000] I believe in myself, I'm really stupid
[02:28.000 --> 02:32.000] I'll show you, hide it, the love is real
[02:32.000 --> 02:35.000] I feel it, it is so dumb
[02:35.000 --> 02:39.000] I give a second chance, I'm cute but
[02:39.000 --> 02:44.000] I believe in myself, I'm really stupid
[02:44.000 --> 02:49.000] I give a second chance, I'm cute but
[02:49.000 --> 02:52.000] I believe in myself
[03:09.000 --> 03:14.000] Thank you for watching!
```

✓ Adding Whisper to a task pipeline

Instead of calling whisper on the command line, we can call it in our python code, so that we can manipulate the input and output with more control.

In this examples, we have a function that takes an audio file path as an input and returns the recognized text (and logs what it thinks the language is).

[Code source](#)

```
# Create a function to process the audio file
# and generate the Whisper output:

def transcribe(audio):

    # load audio and pad/trim it to fit 30 seconds
    audio = whisper.load_audio(audio)
    audio = whisper.pad_or_trim(audio)

    # make log-Mel spectrogram and move to the same device as the model
    mel = whisper.log_mel_spectrogram(audio).to(model.device)

    # detect the spoken language
    _, probs = model.detect_language(mel)
    print(f"Detected language: {max(probs, key=probs.get)}")

    # decode the audio
    options = whisper.DecodingOptions()
    result = whisper.decode(model, mel, options)
    return result.text
```

```
# Transcribe two recording files:
easy_text = transcribe("/content/mary.mp3")
print(easy_text)

hard_text = transcribe("/content/daisy_HAL_9000.mp3")
print(hard_text)
```

```
↔ Detected language: en
Mary had a little lamb, its fleece was white as snow, and everywhere that Mary
Detected language: en
Tazy, Tazy, Tazy. Give me your answer to time after crazy all for the love of
```

✓ Web UI Toolkit for Recording

A simple API for recording and processing audio that uses Gradio.

[Code Source](#)

Note: this may ask for browser permissions, and the recording function may or may not work on Colab.

```
! pip install gradio -q
```

```
↔ _____ 46.5/46.5 MB 24.4 MB/s eta 0:00:00
_____ 322.2/322.2 kB 26.7 MB/s eta 0:00:00
_____ 95.2/95.2 kB 9.1 MB/s eta 0:00:00
_____ 11.3/11.3 MB 78.3 MB/s eta 0:00:00
_____ 72.0/72.0 kB 6.7 MB/s eta 0:00:00
_____ 62.3/62.3 kB 5.8 MB/s eta 0:00:00
```

```
import gradio as gr
import time
```

```
gr.Interface(
    title = 'OpenAI Whisper ASR Gradio Web UI',
    fn=transcribe,
    inputs=[
        gr.Audio(type="filepath")
    ],
    outputs=[
        "textbox"
    ],
)
```

```
live=True).launch()
```

➦ Running Gradio in a Colab notebook requires sharing enabled. Automatically set Colab notebook detected. To show errors in colab notebook, set debug=True in l * Running on public URL: <https://ef10e686f6bd40c51f.gradio.live>

This share link expires in 72 hours. For free permanent hosting and GPU upgrad



No interface is running right now

✓ Visualising and Analysing STT Output

Now that we have text for the audio we processed, we can use it for downstream analysis. Let's try and visualise the data to see what is being talked about.

```
# Sample text
# Read text from file
with open("/content/AllStar_transcription/AllStar.txt", "r", encoding="utf-8") as
    text = file.read().replace("\n", " ") # Remove line breaks
print(text[:100])
```

⇒ Somebody once told me the world is gonna roll me I ain't the sharpest tool in

```
# import libraries
import nltk
from wordcloud import WordCloud
import matplotlib.pyplot as plt
from nltk.corpus import stopwords
import string
import pprint
import re
```

```
# Download list of stopwords. This ensures common words like 'the'
# don't get featured in the wordcloud. This list is for English
nltk.download('stopwords')
```

```
# Load stop words
stop_words = set(stopwords.words('english'))
```

```
# List of English stopwords
stop_words = stopwords.words('english')
pprint.pprint(stop_words, width=150)
```

⇒ ['a',
'about',
'above',
'after',
'again',
'against',
'ain',
'all',
'am',
'an',
'and',
'any',

'are',
'aren',
"aren't",
'as',
'at',
'be',
'because',
'been',
'before',
'being',
'below',
'between',
'both',
'but',
'by',
'can',
'couldn',
"couldn't",
'd',
'did',
'didn',
"didn't",
'do',
'does',
'doesn',
"doesn't",
'doing',
'don',
"don't",
'down',
'during',
'each',
'few',
'for',
'from',
'further',
'had',
'hadn',
"hadn't",
'has',
'hasn',
"hasn't",
'have',
'haven',
"haven't",
'having',
'he'.

```
# Preprocess text with/without stopwords
def preprocess_text(text):
    text = text.lower() # Convert to lowercase
    text = re.sub(r"[^\w\s']", "", text) # Remove punctuation except apostrophes
    words = text.split() # Tokenize
    return " ".join(words)

def remove_stop(text):
    text = text.lower() # Convert to lowercase
    text = re.sub(r"[^\w\s']", "", text) # Remove punctuation except apostrophes
    words = text.split() # Tokenize
    words = [word for word in words if word not in stop_words] # Remove stop words
    return " ".join(words)

# Cleaned text
text_with_stop = preprocess_text(text)
pprint.pprint(text_with_stop, width=50)
```

```
➦ ('somebody once told me the world is gonna roll '
    'me i ain't the sharpest tool in the shed she '
    'was looking kind of dumb with her finger and '
    'her thumb and the shape of an l on her '
    'forehead well the years start coming and they '
    "don't stop coming fed to the rules and i hit "
    "the ground running didn't make sense not to "
    'live for fun your brain gets smart but your '
    'head gets dumb so much to do so much to see so '
    "what's wrong we're taking the back streets "
    "you'll never know if you don't go you'll never "
    "shine if you don't glow hey now you're an "
    'allstar get your game on go play hey now '
    "you're a rock star get the show on get paid "
    "and all that's left in us is gold only "
    "shooting stars break the mold it's a cool "
    "place and they say it gets colder you're "
    'bundled up now wait till you get older but the '
    'media men beg to differ judging by the hole in '
    'the satellite picture the ice we skate is '
    "getting pretty thin the water's getting warm "
    'so you might as well swim but worlds on fire '
    "how about yours that's the way i like it and "
    "i'll never get bored hey now you're an allstar "
    "get your game on go play hey now you're a rock "
    "star get the show on get paid and all that's "
    'left in us is gold only shooting stars break ')
```


"the mold go for the moon set hey now you're an "
'allstar get your game on go play hey now '
"you're a rock star get the show on get paid "
"and all that's left in us is gold only "
'shooting stars somebody once asked could i '
'spare some change well guess i need to get '
'myself away from this place i said yeah what a '
'concept i could use a little fuel myself and '
'we could all use a little change well the '
"years start coming and they don't stop coming "
"fed to the rules that i'll hit the ground "
"running didn't make sense not to live for fun "
'your brain gets smart but your head gets dumb '
"so much to do so much to see so what's wrong "
"we're taking the backstreets you'll never know "
"if you don't go you'll never shine if you "
"don't glow hey now you're an allstar get your "
"game on go play hey now you're a rock star get "
"the show on get paid and all that's left in us "
'is gold only shooting stars break the mold and '
"all that's left in us is gold only shooting "
'stars break the mold')

```
text_without_stop = remove_stop(text)
pprint.pprint(text_without_stop, width=50)
```

```
↔ ("somebody told world gonna roll ain't sharpest "
   'tool shed looking kind dumb finger thumb shape '
   'l forehead well years start coming stop coming '
   'fed rules hit ground running make sense live '
   'fun brain gets smart head gets dumb much much '
   "see what's wrong taking back streets never "
   'know go never shine glow hey allstar get game '
   "go play hey rock star get show get paid that's "
   'left us gold shooting stars break mold cool '
   'place say gets colder bundled wait till get '
   'older media men beg differ judging hole '
   'satellite picture ice skate getting pretty '
   "thin water's getting warm might well swim "
   "worlds fire that's way like never get bored "
   'hey allstar get game go play hey rock star get '
   "show get paid that's left us gold shooting "
   'stars break mold go moon set hey allstar get '
   'game go play hey rock star get show get paid '
   "that's left us gold shooting stars somebody "
   'asked could spare change well guess need get '
   'away place said yeah concept could use little '
   'fuel could use little change well years start '
   'coming stop coming fed rules hit ground '
   'running make sense live fun brain gets smart '
   "head gets dumb much much see what's wrong "
   'taking backstreets never know go never shine '
   'glow hey allstar get game go play hey rock '
   "star get show get paid that's left us gold "
   "shooting stars break mold that's left us gold "
   'shooting stars break mold')
```

```
# Plot the word cloud
plt.figure(figsize=(10, 5))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")
plt.show()
```





✓ Calculating WERs to evaluate your model

In order to understand how well Whisper or any speech model performs on your tasks and dataset, conducting a word error rate analysis may be needed. In this example, we use the `jiwer` python package to test how Whisper performs on the Libri Speech dataset.

This dataset comes with human-generated transcripts (or 'gold transcripts'). For every utterance, we compare the output of Whisper AI with that of the gold transcript, and aggregate the scores over the dataset.

Note: this will take some compute time [Code Source](#)

```
# check your specs
model.device
```

```
➞ device(type='cuda', index=0)
```

```
! pip install git+https://github.com/openai/whisper.git
! pip install jiwer
```

```
➞ Collecting git+https://github.com/openai/whisper.git
  Cloning https://github.com/openai/whisper.git to /tmp/pip-req-build-orm2utq
  Running command git clone --filter=blob:none --quiet https://github.com/openai/whisper.git
  Resolved https://github.com/openai/whisper.git to commit 517a43ecd132a2089d8
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: more-itertools in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: numba in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: tiktoken in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: torch in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: triton>=2 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: llvmlite<0.44,>=0.43.0dev0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: regex>=2022.1.18 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: requests>=2.26.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: fsspec in /usr/local/lib/python3.11/dist-packages
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local,
```

```

Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages/nvidia/cuda_runtime-12.4.127-cp311-cp311-manylinux_2_17_x86_64.whl (19.1 MB)
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages/nvidia/cuda_cupti-12.4.127-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: nvidia-cudnn-cu12==9.1.0.70 in /usr/local/lib/python3.11/dist-packages/nvidia/cudnn-9.1.0.70-cp311-cp311-manylinux_2_17_x86_64.whl (660.8 MB)
Requirement already satisfied: nvidia-cublas-cu12==12.4.5.8 in /usr/local/lib/python3.11/dist-packages/nvidia/cublas-12.4.5.8-cp311-cp311-manylinux_2_17_x86_64.whl (363.6 MB)
Requirement already satisfied: nvidia-cufft-cu12==11.2.1.3 in /usr/local/lib/python3.11/dist-packages/nvidia/cufft-11.2.1.3-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: nvidia-curand-cu12==10.3.5.147 in /usr/local/lib/python3.11/dist-packages/nvidia/curand-10.3.5.147-cp311-cp311-manylinux_2_17_x86_64.whl (20.1 MB)
Requirement already satisfied: nvidia-cusolver-cu12==11.6.1.9 in /usr/local/lib/python3.11/dist-packages/nvidia/cusolver-11.6.1.9-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in /usr/local/lib/python3.11/dist-packages/nvidia/cusparselt-0.6.2-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in /usr/local/lib/python3.11/dist-packages/nvidia/nccl-2.21.5-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages/nvidia/nvtx-12.4.127-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: nvidia-nvjitlink-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages/nvidia/nvjitlink-12.4.127-cp311-cp311-manylinux_2_17_x86_64.whl (1.2 MB)
Requirement already satisfied: sympy==1.13.1 in /usr/local/lib/python3.11/dist-packages/sympy-1.13.1-py3-none-any.whl (3.8 MB)
Requirement already satisfied: mpmath<1.4, >=1.1.0 in /usr/local/lib/python3.11/dist-packages/mpmath-1.3.0-py3-none-any.whl (536 kB)
Requirement already satisfied: charset-normalizer<4, >=2 in /usr/local/lib/python3.11/dist-packages/charset-normalizer-3.3.0-py3-none-any.whl (65 kB)
Requirement already satisfied: idna<4, >=2.5 in /usr/local/lib/python3.11/dist-packages/idna-3.10-py3-none-any.whl (70 kB)
Requirement already satisfied: urllib3<3, >=1.21.1 in /usr/local/lib/python3.11/dist-packages/urllib3-2.2.1-py3-none-any.whl (121 kB)
Requirement already satisfied: certifi<2025.1.1, >=2017.4.17 in /usr/local/lib/python3.11/dist-packages/certifi-2024.12.14-py3-none-any.whl (110 kB)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages/MarkupSafe-2.1.5-cp311-cp311-manylinux_2_17_x86_64.whl (16 kB)
Collecting jiwer
  Downloading jiwer-3.1.0-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: click>=8.1.8 in /usr/local/lib/python3.11/dist-packages/click-8.1.8-py3-none-any.whl (98 kB)
Collecting rapidfuzz>=3.9.7 (from jiwer)
  Downloading rapidfuzz-3.12.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1/3.1 MB)
  Downloading jiwer-3.1.0-py3-none-any.whl (22 kB)
  Downloading rapidfuzz-3.12.2-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1/3.1 MB)
  3.1/3.1 MB 42.1 MB/s eta 0:00:00
Installing collected packages: rapidfuzz, jiwer
Successfully installed jiwer-3.1.0 rapidfuzz-3.12.2

```

```
#Setup libraries and check for errors
import os
import numpy as np

try:
    import tensorflow # required in Colab to avoid protobuf compatibility issues
except ImportError:
    pass

import torch
import pandas as pd
import whisper
import torchaudio

from tqdm.notebook import tqdm

DEVICE = "cuda" if torch.cuda.is_available() else "cpu"

print(DEVICE)

🔄 cuda
```

```
# A python class to process the dataset, and return the audio data Whisper needs,
# with the gold transcripts per file.
# Note: This only uses the first 30 seconds, for efficiency
```

```
class LibriSpeech(torch.utils.data.Dataset):
    """
    A simple class to wrap LibriSpeech and trim/pad the audio to 30 seconds.
    It will drop the last few seconds of a very small portion of the utterances.
    """
    def __init__(self, split="test-clean", device=DEVICE):
        self.dataset = torchaudio.datasets.LIBRISPEECH(
            root=os.path.expanduser("~/cache"),
            url=split,
            download=True,
        )
        self.device = device

    def __len__(self):
        return len(self.dataset)

    def __getitem__(self, item):
        audio, sample_rate, text, _, _, _ = self.dataset[item]
        assert sample_rate == 16000
        audio = whisper.pad_or_trim(audio.flatten()).to(self.device)
        mel_spectrogram = whisper.log_mel_spectrogram(audio)

        return (mel_spectrogram, text)

# Set up the Dataset:
libriSpeech_dataset = LibriSpeech("test-clean")
libriSpeech_dataset_torchDataLoader = torch.utils.data.DataLoader(libriSpeech_dataset)
```

➡ 100%|██████████| 331M/331M [00:50<00:00, 6.87MB/s]


```


[⇒] 100%|████████████████████████████████████████| 139M/139M [00:38<00:00, 3.77MiB,
Model is English-only and has 71,825,408 parameters.

```

100% 164/164 [02:06<00:00, 1.19it/s]

```
# Display the generated data:
# Adjust pandas display options to show the entire text in a column
pd.set_option('display.max_colwidth', None)

# Save transcripts to a table
data = pd.DataFrame(dict(generated_transcripts=generated_transcripts, gold_transcripts=gold_transcripts))
```




| | generated_transcripts | gold_transcripts |
|------|---|--|
| 0 | He hoped there would be stew for dinner, turnips and carrots and bruised potatoes and fat mutton pieces to be ladled out in thick, peppered flower-faten sauce. | HE HOPED THERE WOULD BE STEW FOR DINNER TURNIPS AND CARROTS AND BRUISED POTATOES AND FAT MUTTON PIECES TO BE LADLED OUT IN THICK PEPPERED FLOUR FATTENED SAUCE |
| 1 | Stuffered into you, his belly counseled him. | STUFF IT INTO YOU HIS BELLY COUNSELLED HIM |
| 2 | After early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels. | AFTER EARLY NIGHTFALL THE YELLOW LAMPS WOULD LIGHT UP HERE AND THERE THE SQUALID QUARTER OF THE BROTHELS |
| 3 | Hello Bertie, any good in your mind? | HELLO BERTIE ANY GOOD IN YOUR MIND |
| 4 | Number 10. Fresh Nelly is waiting on you. Good night, husband. | NUMBER TEN FRESH NELLY IS WAITING ON YOU GOOD NIGHT HUSBAND |
| ... | ... | ... |
| 2615 | Oh, to shoot my soul's full meaning into future years, that they should lend it utterance and salute love that endures from life that disappears. | OH TO SHOOT MY SOUL'S FULL MEANING INTO FUTURE YEARS THAT THEY SHOULD LEND IT UTTERANCE AND SALUTE LOVE THAT ENDURES FROM LIFE THAT DISAPPEARS |
| 2616 | Then I, long tried by natural ills, received the comfort fast. While budding, at thy sight my pilgrim's staff | THEN I LONG TRIED BY NATURAL ILLS RECEIVED THE COMFORT FAST WHILE BUDDING AT THY SIGHT MY PILGRIM'S STAFF |

```
# Normalize the data (change case, remove special characters, etc)
```

```
import jiwer
from whisper.normalizers import EnglishTextNormalizer

normalizer = EnglishTextNormalizer()
```

```
data["generated_transcripts_clean"] = [normalizer(text) for text in data["generated_transcripts"]]
data["gold_transcripts_clean"] = [normalizer(text) for text in data["gold_transcripts"]]
data
```



| | generated_transcripts | gold_transcripts | generated_transcripts_clean | gold_transcripts_clean |
|-----|---|--|--|------------------------|
| 0 | He hoped there would be stew for dinner, turnips and carrots and bruised potatoes and fat mutton pieces to be ladled out in thick, peppered flower-faten sauce. | HE HOPED THERE WOULD BE STEW FOR DINNER TURNIPS AND CARROTS AND BRUISED POTATOES AND FAT MUTTON PIECES TO BE LADLED OUT IN THICK PEPPERED FLOUR FATTENED SAUCE | he hoped there would be stew for dinner turnips and carrots and bruised potatoes and fat mutton pieces to be ladled out in thick peppered flower faten sauce | ca e le |
| 1 | Stuffered into you, his belly counseled him. | STUFF IT INTO YOU HIS BELLY COUNSELLED HIM | stuffered into you his belly counseled him | |
| 2 | After early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels. | AFTER EARLY NIGHTFALL THE YELLOW LAMPS WOULD LIGHT UP HERE AND THERE THE SQUALID QUARTER OF THE BROTHEL | after early nightfall the yellow lamps would light up here and there the squalid quarter of the brothels | aft an |
| 3 | Hello Bertie, any good in your mind? | HELLO BERTIE ANY GOOD IN YOUR MIND | hello bertie any good in your mind | he |
| 4 | Number 10. Fresh Nelly is waiting on you. Good night, husband. | NUMBER TEN FRESH NELLY IS WAITING ON YOU GOOD NIGHT HUSBAND | number 10 fresh nelly is waiting on you good night husband | |
| ... | ... | ... | ... | ... |
| | | OH TO SHOOT MY | | |

```
# Calculate the word error rate for the dataset:
```

```
wer = jiwer.wer(list(data["gold_transcripts_clean"]), list(data["generated_transc  
print(f"WER: {wer * 100:.2f} %")
```

```
↔ WER: 4.27 %
```

For Whisper, tested on 2620 audio files, the average errors in transcription are % 4.27. For your project, you can assess if this is something you can live with, or you can decide to choose a different transcription system.

✓ References and Further Reading

- [Illustrated Wav2vec 2.0](#)
- [Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations](#)
- [Robust Speech Recognition via Large-Scale Weak Supervision](#)

```
pip install nbstripout  
jupyter nbstripout Speech_to_Text_with_Whisper.ipynb
```

