Whisper Transcription API Documentation

Version: 1.4
Device: CUDA / CPU
Framework: Flask + Whisper + WebRTC VAD

Overview

This project provides a plug-and-play Speech-to-Text HTTP API using OpenAI's Whisper model, enhanced with advanced filtering techniques and silence detection (VAD).

It is specifically designed to improve transcription accuracy in noisy, silent, or low-quality audio by filtering out irrelevant or low-confidence text before returning results.

✅ What This API Does

- Accepts audio input (WAV) via HTTP POST request
- Uses OpenAI's Whisper model for speech recognition
- Performs voice activity detection (VAD)
- Filters hallucinated, repetitive, and low-confidence transcriptions
- Returns clean, structured JSON output with final text and segment metadata

---

Project Structure

File	Description
`app.py`	Main Python script running the Flask API
`requirements.txt`	List of dependencies required to run the project
`README.md`	This documentation file

Setup & Installation Guide

Python

- Install Python 3.8 or higher from python.org

FFMPEG Setup

FFMPEG is required by `pydub` and Whisper for audio processing.

Ubuntu:

sudo apt install ffmpeg

Mac (Homebrew):

brew install ffmpeg

Windows:

1. Download a static build from: FFmpeg Downloads
2. Extract it and copy the `bin` folder path (e.g., `C:\ffmpeg\bin`)
3. Add it to your system's PATH via **Environment Variables**
4. Confirm with:

ffmpeg -version

⚙️ Build Tools & System Requirements

- **Ubuntu/Debian:**
  sudo apt update && sudo apt install ffmpeg build-essential python3-dev -y
- **Windows:**
  1. Install FFmpeg and add it to your PATH
  2. Install Microsoft Visual C++ Build Tools
     - During installation, select:
       - ☑ MSVC v143 - VS 2022 C++ x64/x86 build tools
       - ☑ Windows 10/11 SDK (10.0.19041.0)
       - ☑ C++ CMake tools for Windows

VAD Setup

`webrtcvad` requires a C compiler. Ensure:

- Python headers are installed (e.g., `python3-dev`)
- On Windows, C++ redistributables and `Build Tools` must be pre-installed

Install VAD:

pip install webrtcvad

Dependencies

Install required Python packages:

pip install -r requirements.txt

Sample `requirements.txt`:

```
flask
webrtcvad
pydub
torch
openai-whisper
setuptools-rust
numba
numpy
tqdm
more-itertools
Tiktoken
pyaudio

# Triton for Linux x86_64 only
triton>=2.0.0; platform_machine == "x86_64" and (sys_platform == "linux" or sys_platform == "linux2")
```

Run the API

python app.py

This will launch the API server on `http://localhost:8001/`

Base URL

http://<host>:8001/

Endpoints

1. `/version`

Method: GET
Description: Returns the application, Whisper, and device version.

✅ Sample Response

```
{
  "app_version": "1.3a",
  "whisper_version": "20240930",
  "device": "cuda"
}
```

---

2. `/model`

Method: GET
Description: Returns the last used Whisper model and device.

✅ Sample Response

```
{
  "model": "large-v3-turbo",
  "device": "cuda"
}
```

---

3. `/transcribe`

Method: POST
Description: Upload an audio file and receive a filtered transcription result.

📂 Form Data Parameters

Parameter	Type	Required	Default	Description
`audio`	file	✅ Yes	-	Audio file (WAV or compatible)
`model`	string	No	`large-v3-turbo`	Whisper model name
`enable_filtering`	bool	No	`false`	Allows complete bypass of filtering logic for segment evaluation.
`avg_logprob_threshold`	float	No	`-1.0`	Filter for average log probability
`compression_ratio_threshold`	float	No	`2.4`	Compression ratio threshold
`no_speech_prob_threshold`	float	No	`0.6`	Silence detection threshold
`temperature`	float	No	`0.0`	Decoding temperature
`vad_aggressiveness`	int	No	`2`	VAD sensitivity (0-3)
`vad_voice_ratio_threshold`	float	No	`0.1`	Ratio of voiced frames to trigger voice detection
`min_text_length`	int	No	`5`	Minimum accepted text length
`wrap_length`	int	No	`32`	Wrapped segment text with configurable line breaks
`enable_vad`	bool	No	`true`	Enable or disable VAD check
`request_id`	string	No	`-`	Custom transcription ID

✅ Sample cURL

```
curl -X POST http://localhost:8001/transcribe \
  -F "audio=@sample.wav" \
  -F "model=base" \
  -F "avg_logprob_threshold=-0.5" \
  -F "compression_ratio_threshold=2.0" \
  -F "no_speech_prob_threshold=0.4" \
  -F "min_text_length=5" \
  -F "enable_vad=true" \
  -F "request_id=my-job-123"
```

✅ Sample Response


```
{
  "antix": {
    "request _id": "my-job-123",
    "api_ver": "1.3",
    "whisper_ver": "20240930",
    "model": "base",
    "device": "cuda",
    "response_time": 3.421,
    “enable_filtering”: True
  },
  "result": {
    "language": "en",
    "segments": [
      {
        "start": 0.0,
        "end": 5.0,
        "text": "Hello world",
        "avg_logprob": -0.3,
        "no_speech_prob": 0.2,
        "compression_ratio": 1.8,
        "antix": {
            "wrapped_text": "You have a moment where you can\ngo forward or you can give up.",
          "filtered": 0,
          "filtered_bin": "0b00000"
        }
      },
      {
        "start": 5.0,
        "end": 10.0,
        "text": "umm...",
        "avg_logprob": -1.2,
        "no_speech_prob": 0.5,
        "compression_ratio": 2.6,
        "antix": {
            "wrapped_text": "You have, You have., You have., You have.",
          "filtered": 14,
          "filtered_bin": "0b01110"
```

```
      }
    }
  ]
 }
}
```

Filter Bitmask Reference

Bit	Binary	Meaning
0	`0b00001`	❌ VAD: No voice detected
1	`0b00010`	❌ Avg log prob too low
2	`0b00100`	❌ Compression too high
3	`0b01000`	❌ No-speech prob too high
4	`0b10000`	❌ Text too short
All passed	`0b00000`	✅ Passed all checks

Supported Whisper Models

tiny, tiny.en, base, base.en, small, small.en, medium, medium.en, large, large-v1, large-v2, large-v3, large-v3-turbo

---

Error Handling

Code	Reason
400	Missing file or invalid form input
400	Invalid parameter or threshold values
200	Transcription returned successfully (check `filtered` flag)

Notes

- Best practice: Use `.wav` mono, 16kHz input
- Disable VAD for music or mixed audio inputs
- Use bitmask (`filtered_bin`) to evaluate segment filtering reason