**Installing and Using WhisperX on a MacOS Platform**

Bigya Bajracharya, Mofeoluwa Jide-Jegede, and Phil Pfeiffer

Last updated 15 March 2024

Contents

[1 Installing Whisper X 1](#_Toc160933920)

[1.1 Download Python 1](#_Toc160933921)

[1.2 Install pip for Python3 1](#_Toc160933922)

[1.3 Install Virtual Environment 1](#_Toc160933923)

[1.4 Create and Activate a Virtual Environment 2](#_Toc160933924)

[1.5 Install WhisperX Dependencies 2](#_Toc160933925)

[1.6 Install WhisperX 2](#_Toc160933926)

[2 Running WhisperX 2](#_Toc160933927)

[3 Running WhisperX with Diarization 2](#_Toc160933928)

[3.1 configuring whisperX for diarization 2](#_Toc160933929)

[3.2 running whisperX with diarization 3](#_Toc160933930)

[4 Exiting the virtual environment 3](#_Toc160933931)

[5 Troubleshooting the installation 3](#_Toc160933932)

[5.1 WhisperX fails with the error message “missing the required positional arguments” 3](#_Toc160933933)

[5.2 Audio file can’t be found, regardless of where it’s positioned 3](#_Toc160933934)

[5.3 Module ffmpeg has no attribute ‘Error’ 3](#_Toc160933935)

[5.4 Value Error 4](#_Toc160933936)

[6 whisperX – Additional Documentation 4](#_Toc160933937)

# Installing Whisper X

Before using WhisperX, use the following steps to set up your environment and install the necessary dependencies.

## Download Python

Download the latest version of Python for Mac OSX from <https://www.python.org/downloads/>. This document assumes that the latest version is 3.12.x. Adjust the instructions accordingly for newer versions of Python.

## Install pip for Python3

1. Open your terminal.
2. Install pip and pip3 for Python3 using the following command:

sudo apt-get install python3-pip

## Install Virtual Environment

We'll use virtualenvwrapper to manage virtual environments.

1. Install virtualenvwrapper by running

sudo pip3 install virtualenvwrapper

You may be asked to enter your administrator password.

1. If you’re using the bash command line interpreter, open your .bashrc file for editing:

nano ~/.bashrc

Else, if you’re using the zsh command line interpreter, open your .zprofile file for editing:

nano ~/.zprofile

1. Add the following lines to your .bashrc or .zprofile file:

export WORKON\_HOME=$HOME/.virtualenvs

export PROJECT\_HOME=$HOME/Devel

export VIRTUALENVWRAPPER\_PYTHON=$(which python3)

source /Library/Frameworks/Python.framework/Versions/3.11/bin/virtualenvwrapper.sh

In the command above, replace /Library/Frameworks/Python.framework/Versions/3.11/bin with the path where virtualenvwrapper.sh is installed. You can find this path by running

which virtualenvwrapper.sh

1. Save the file. Press `Ctrl+X` then confirm with 'Y' to exit.
2. Execute the updated .bashrc or .zprofile file:

source ~/.bashrc

or

source ~/.zprofile

If you encounter the error **virtualenvwrapper.sh: There was a problem running the initialization hooks**, use the following command to set the VIRTUALENVWRAPPER\_PYTHON environment variable:

export VIRTUALENVWRAPPER\_PYTHON=$(which python3)

## Create and Activate a Virtual Environment

1. Create a new virtual environment with your desired name (e.g., "temp"):

mkvirtualenv temp

This command creates and activates a new environment named "temp."

1. Check if the environment has been created:

workon

This command should display "temp". If you encounter the **zsh: command not found: workon**  error, repeat steps 3.3 to 3.7 to configure virtualenvwrapper for your shell.

## Install WhisperX Dependencies

1. Switch to the virtual environment you created (e.g., "temp"):

workon temp

1. Install WhisperX dependencies:

pip3 install ffmpeg-python

pip3 install torch==2.0.0 torchaudio==2.0.0

## Install WhisperX

Install WhisperX using pip:

pip3 install git+https://github.com/m-bain/whisperx.git

# Running WhisperX

To use WhisperX to transcribe audio, run the following command, replacing "audio.wav" with the path to your audio file:

whisperx audio.wav

# Running WhisperX with Diarization

## configuring whisperX for diarization

1. Create a Hugging Face Account
   1. Go to [Hugging Face Join Page](https://huggingface.co/join).
   2. Specify a unique username and a secure password during the registration process.

This account is essential for obtaining access to the WhisperX pipelines that support diarization.

1. Agree to Terms and Conditions

Navigate to the following libraries on Hugging Face and agree to their terms and conditions, as WhisperX uses these libraries for diarization:

* 1. [Hugging Face Segmentation Library](https://huggingface.co/pyannote/segmentation): Follow the provided instructions.
  2. [Hugging Face Voice Activity Detection Library](https://huggingface.co/pyannote/voice-activity-detection): Follow the provided instructions.
  3. Hugging [Face](https://huggingface.co/pyannote/speaker-diarization-3.1) Speaker Diarization: Follow the provided directions.

Some of these steps will ask for an organization and a website.

1. Generate an Access Token

Create a token to access Hugging Face software from your host platform.

* 1. Browse to <https://huggingface.co/settings/tokens>
  2. Follow directions

This token will be used as the `hf\_token` parameter for diarization.

Also recommended: install support for NVIDIA CUDA on hardware platforms that support GPUs.

* For the NVIDIA CUDA drivers, along with documentation for how to install them, see <https://developer.nvidia.com/cuda-downloads>.
* In order to enable WhisperX’s use of CUDA devices, follow the directions at <https://stackoverflow.com/questions/75775272/cuda-and-openai-whisper-enforcing-gpu-instead-of-cpu-not-working>:
  + Configure torch to use CUDA: i.e., execute

**python -m pip install torch torchvision torchaudio --index-url** [**https://download.pytorch.org/whl/cu118**](https://download.pytorch.org/whl/cu118)

* + Reconfigure WhisperX to use the updated torch: i.e., execute

**python -m pip install -U openai-whisper**

* + Confirm that torch has been configured to use CUDA.
    - Open a python interpreter
    - At the prompt, execute **import torch**
    - After this command completes—it may take some time—execute  **torch.cuda.is\_available()** This last command should return **True.**

While the CUDA argument can be passed as the model’s device parameter, according to this post’s authors, WhisperX will automatically attempt to use a GPU if it detects one on the host platform.

## running whisperX with diarization

1. Open your terminal
2. Run the following command to transcribe and diarize your file:

whisperx audio.wav --hf\_token <token> --diarize

* Replace `audio.wav` with the audio file you want to perform diarization on.
* Replace `<token>` with the token you generated in step 3.

# Exiting the virtual environment

To exit the virtual environment, issue the command **deactivate**.

# Troubleshooting the installation

## WhisperX fails with the error message “missing the required positional arguments”

This is a problem in WhisperX version 3.1.2. The syndrome is as follows:

TypeError: **new**() missing 3 required positional arguments: 'max\_new\_tokens', 'clip\_timestamps', and 'hallucination\_silence\_threshold'

To fix this error, after installing WhisperX,

* Locate the file **asr.py** in the WhisperX distribution. It should be in the WhisperX package home directory.
* Ensure the code immediately after **"suppress\_numerals": False** (ca. line 322) has the following three lines. If not, add them:

"**max\_new\_tokens": None,**

**"clip\_timestamps": None,**

**"hallucination\_silence\_threshold": None,**

Alternatively, reload **asr.py** from the PiPy repository; the problem has supposedly been finsed.

## Audio file can’t be found, regardless of where it’s positioned

WhisperX’s load\_**audio()** routine (**audio.py**) raises **RuntimeError(‘Failed to load audio’)** without first checking if ffmpeg.exe has been installed. Verify that **ffmpeg.exe** has been properly installed.

## Module ffmpeg has no attribute ‘Error’

If you encounter **AttributeError: module 'ffmpeg' has no attribute 'Error'** try reinstalling ffmpeg, as follows:

**pip3 uninstall ffmpeg**

**pip3 uninstall ffmpeg-python**

**pip3 install ffmpeg-python**

## Value Error

If you encounter **ValueError: Requested float16 compute type...,** use the following command to specify the compute type as int8:

whisperx audio.wav --compute\_type int8

# whisperX – Additional Documentation

A list of whisperX options can be obtained by running **whisperx --help.**  Documentation for the five whisperX models—tiny, base, small, medium, and large—can be obtained from <https://github.com/openai/whisper/blob/main/README.md>

For more details and troubleshooting, refer to the WhisperX documentation at **https://github.com/m-bain/whisperX#setup-%EF%B8%8F**