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

Bigya Bajracharya, Mofeoluwa Jide-Jegede, and Phil Pfeiffer

Last updated 15 March 2024

Contents

[1 Installing WhisperX 1](#_Toc160934297)

[1.1 Download Python 1](#_Toc160934298)

[1.2 Install pip for Python3 1](#_Toc160934299)

[1.3 Install Virtual Environment 1](#_Toc160934300)

[1.4 Create and Activate a Virtual Environment 2](#_Toc160934301)

[1.5 Install WhisperX Dependencies 2](#_Toc160934302)

[1.6 Install WhisperX 2](#_Toc160934303)

[2 Using WhisperX to transcribe an audio file 2](#_Toc160934304)

[3 Using WhisperX to transcribe and diarize an audio file 3](#_Toc160934305)

[3.1 configuring WhisperX for supporting diarization 3](#_Toc160934306)

[3.2 running WhisperX with diarization 3](#_Toc160934307)

[4 Exiting the virtual environment 3](#_Toc160934308)

[5 Troubleshooting the installation 3](#_Toc160934309)

[5.1 WhisperX fails with the error message “missing the required positional arguments” 3](#_Toc160934310)

[5.2 Audio file can’t be found, regardless of where it’s positioned 4](#_Toc160934311)

[5.3 Module ffmpeg has no attribute ‘Error’ 4](#_Toc160934312)

[5.4 Value Error 4](#_Toc160934313)

[6 whisperX – Additional Documentation 4](#_Toc160934314)

# Installing WhisperX

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 Linux from <https://www.python.org/downloads/> This document assumes that the latest version is 3.11.x. Adjust the instructions accordingly for newer versions of Python.

## Install pip for Python3

1. Open your terminal.
2. Download get-pip.py from https://bootstrap.pypa.io/get-pip.py
3. Install pip and pip3 for Python3 using the following command:

sudo python3 get-pip.py

## 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 was installed. You can find this path by running

which virtualenvwrapper.sh!nwopipe

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:

sudo pip3 install ffmpeg-python2

sudo pip3 install torch==2.0.0 torchaudio==2.0.2

## Install WhisperX

1. Install WhisperX using pip:

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

Notes:

* If the installation process reports that the most recent version of a supporting package must be downgraded because of an incompatibility with the current version of whisperx, use pip to downgrade the package. For example, to downgrade protobuf to version 3.20, use the following command:

sudo pip3 install --force-reinstall protobuf==3.20

1. Use the following command to confirm that WhisperX installed properly:

whisperx --help

# Using WhisperX to transcribe an audio file

At the command prompt, run the command **whisperx *<path><filename>* --compute\_type int8** where ***<filename>*** is the name of the file to transcribe and ***<path>*** is the file system path to that file.

* For example, enter **whisperx recordings/audio001.mp3 --compute\_type int8** when
* the file is in the current directory’s recordings subdirectory
* the file name is “audio001.mp3.”
* Note that an additional slash follows the file path then the audio file name.

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

If you encounter **AttributeError: module 'ffmpeg' has no attribute 'Error'** try the following:

**pip3 uninstall ffmpeg**

**pip3 uninstall ffmpeg-python**

**pip3 install ffmpeg-python**

# Using WhisperX to transcribe and diarize an audio file

## configuring WhisperX for supporting diarization

I’m not sure what all steps in the following sequence of steps were essential to adding support for diarization to whisperX. What I do know is that I—Phil—worked at this for four days before figuring it all out. Accordingly, I recommend that someone try experimenting with the Windows CLI part of this process to see if some of these steps can be eliminated.

1. Establish an account on huggingface.co
   1. Browse to https://huggingface.co/join
   2. Specify a user name and a password

* This account is needed to obtain access to the whisperX “pipelines” for supporting diarization

1. Agree to the terms and conditions that govern access to the three libraries that whisperX uses for diarization
   1. Browse to <https://huggingface.co/pyannote/segmentation>. Follow directions.
   2. Browse to <https://huggingface.co/pyannote/voice-activity-detection> Follow directions.
   3. Browse to <https://huggingface.co/pyannote/speaker-diarization-3.1>. Follow directions.

* You will need to specify an organization and a website as a part of these processes. I’m not sure how carefully this requirement is checked.

1. Generate a token for accessing HuggingFace software from your host platform.
   1. Browse to <https://huggingface.co/settings/tokens>
   2. Follow directions

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

Run whisperx as before, with the additiional option **–diarize --hf *<key>***, where ***<key>*** is the token you obtained from <https://huggingface.co>.

For more information on whisperX, see https://github.com/m-bain/whisperx .

# 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**