**Experiment 4: Perform Speech-to-text conversion using pre trained machine learning API models for short-form or long-form audio.**

**Requirement:** Laptop or Desktop with Python installed

**Theory**

Far from a being a fad, the overwhelming success of speech-enabled products like Amazon Alexa has proven that some degree of speech support will be an essential aspect of household tech for the foreseeable future. If you think about it, the reasons why are pretty obvious. Incorporating speech recognition into your Python application offers a level of interactivity and accessibility that few technologies can match.

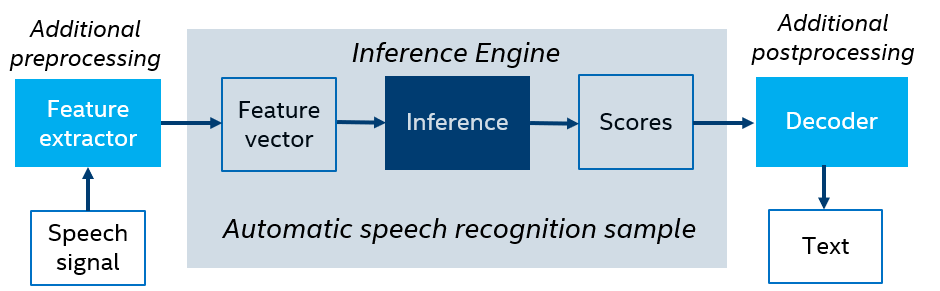
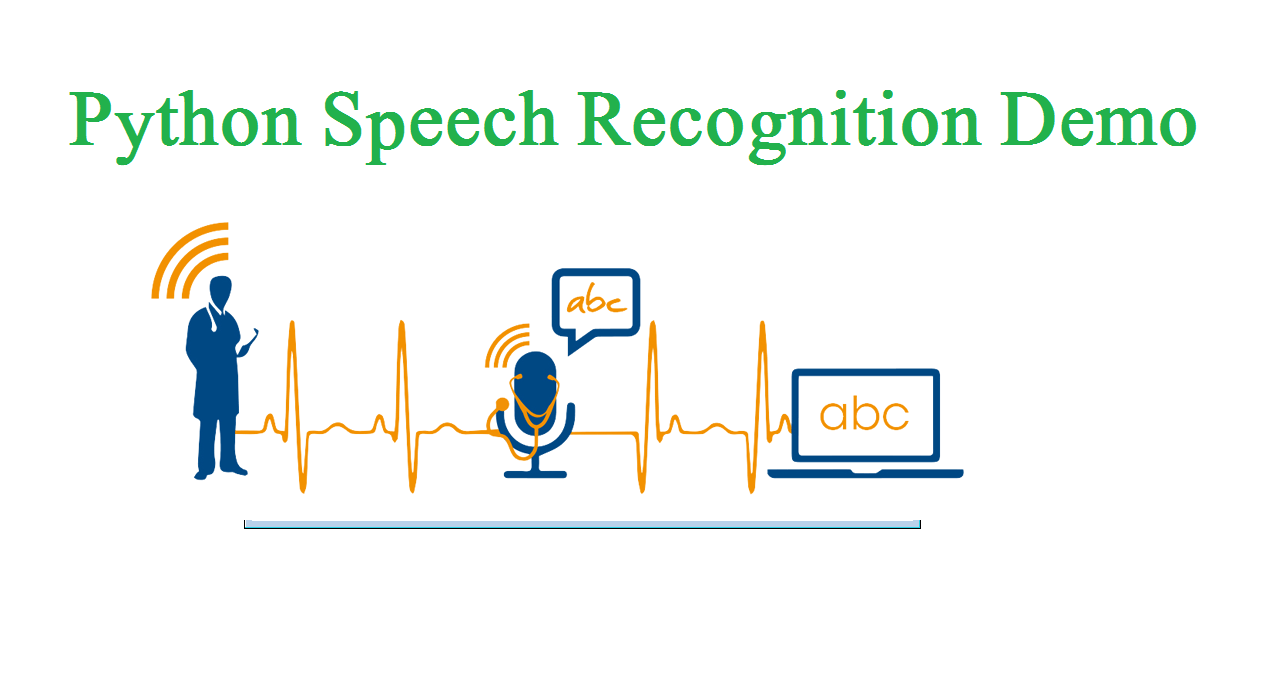


Figure 1: Speech recognition in Python

Best of all, including speech recognition in a Python project is really simple. In this guide, you’ll find out how. You’ll learn:

* How speech recognition works,
* What packages are available on Python and
* How to install and use the Speech Recognition package—a full-featured and easy-to-use Python speech recognition library.

The first component of speech recognition is, of course, speech. Speech must be converted from physical sound to an electrical signal with a microphone/mike, and then to digital data with an analog-to-digital converter(A/D converter). Once digitized, several models can be used to transcribe the audio to text.

Most modern speech recognition systems rely on what is known as a Hidden Markov Model (HMM). This approach works on the assumption that a speech signal, when viewed on a short enough timescale (say, ten milliseconds), can be reasonably approximated as a stationary process—that is, a process in which statistical properties do not change over time.

In a typical HMM, the speech signal is divided into 10-millisecond fragments. The power spectrum of each fragment, which is essentially a plot of the signal’s power as a function of frequency, is mapped to a vector of real numbers known as cepstral coefficients. The dimension of this vector is usually small—sometimes as low as 10, although more accurate systems may have dimension 32 or more. The final output of the HMM is a sequence of these vectors.

To decode the speech into text, groups of vectors are matched to one or more phonemes—a fundamental unit of speech. This calculation requires training, since the sound of a phoneme varies from speaker to speaker, and even varies from one utterance to another by the same speaker. A special algorithm is then applied to determine the most likely word (or words) that produce the given sequence of phonemes.

One can imagine that this whole process may be computationally expensive. In many modern speech recognition systems, neural networks are used to simplify the speech signal using techniques for feature transformation and dimensionality reduction before HMM recognition. Voice activity detectors (VADs) are also used to reduce an audio signal to only the portions that are likely to contain speech. This prevents the recognizer from wasting time analyzing unnecessary parts of the signal.

Fortunately, as a Python programmer, you don’t have to worry about any of this. A number of speech recognition services are available for use online through an API, and many of these services offer Python SDKs.

# Convert Speech to text and text to Speech

# Install “PyAudio” library

# pip install pipwin

# pipwin install pyaudio

# Note If you get error in Step 1 than follow step 2

# If you are using python 3.7 then use the below one

# For 32-bit

# pip install PyAudio-0.2.11-cp37-cp37m-win32.whl

**For 64-bit**

**pip install PyAudio-0.2.11-cp37-cp37m-win\_amd64.whl**

# If you are using python 3.8 then use the below one

# pip install PyAudio-0.2.11-cp38-cp38-win\_amd64.whl

# Install “Speech Recognition” library

# pip install SpeechRecognition

# Install “Natural Language Toolkit”

# pip install nltk

# Install “textblob” library

# pip install textblob

# Install “NLTK Data”

# python -m textblob.download\_corpora

# Python pyttsx3 module:

# pip install pyttsx3

# Install gTTs module:

# pip install gtts

# Copy the following two files from the folder

# (C:\Users\upadh\AppData\Local\Programs\Python\Python38\Lib\site-packages\pywin32\_system32)

# pythoncom38.dll

# pywintypes38.dll

# to (C:\Users\upadh\AppData\Local\Programs\Python\Python38)

**Output on Next Page**

**Output**

