

- *Pocketsphinx* is an open source speech decoder developed under the CMU Sphinx Project. It is quite fast and has been designed to work well on mobile operating systems such as Android as well as embedded systems (like Raspberry Pi). The advantage of using Pocketsphinx is that the speech recognition is performed offline, which means you don't need an active Internet connection. However, the recognition rate is nowhere close to that of Google's STT.
- *AT&T STT* was developed by AT&T. The recognition rate is good, but it needs an active connection to work, just like Google STT.
- *Julius* is a high-performance, open source speech-recognition engine. It does not need an active Internet connection, like Pocketsphinx. It is quite complicated to use because it requires the user to train their own acoustic models.
- *Wit.ai STT* is a cloud-based service provided to users. Like AT&T and Google STT, it requires an active Internet connection to work.
- *IBM STT* was developed by IBM and is a part of the Watson division. It requires an active Internet connection to work.

This project uses Google STT because it is one of the most accurate STT engines available. In order to use Google STT in your project, you need a Python module called `SpeechRecognition`.

Installing SpeechRecognition

You install `SpeechRecognition` by issuing the following command via the terminal:

```
$ pip install SpeechRecognition
```

This sets up the `SpeechRecognition` module for you. This library supports Google Speech Recognition, Wit.ai, IBM Speech to Text, and AT&T Speech to Text. You can choose any of these for your version of Melissa.

Recording Audio to a WAV File

Let's write a small Python program to see how this library works. This program records the user's voice and saves it to a `.wav` file. Recording the audio to a WAV file will help you get comfortable with the `SpeechRecognition` library. You also use this method of recording speech to a WAV file and then passing that file to the STT server in Chapter 8:

```
import speech_recognition as sr

r = sr.Recognizer()
with sr.Microphone() as source:
```