Survey on Voice Recognition Software for Robotics

Software and Libraries Reviewed

1. Tensorflow
2. Google Cloud Speech
3. Kaldi (pykaldi)
4. SpeechRecognition
5. CMUSphinx pocketsphinx

A brief introduction

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, can be reasonably approximated as a stationary process.

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 (basically the result of taking an inverse Fourier Transform of the power spectrum of the signal). 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* — the unit of speech. This calculation requires training, since such pronunciation varies from speaker to speaker and much more. This is an example of a classification algorithm.

Case 1: Tensorflow

Tensorflow is an open source machine learning library. It doesn't provide a Speech Recognition API, but rather a suit of Machine Learning tools that you *can* use to create your own Speech Recognizer. Tutorials such as (1) are of help in this task. However, such examples are too limited in their reach (they can help recognize set of 10-20 words but hours of training and quite big databases). For the present project, I believe it's not a convenient choice unless we require to train a speech model from scratch. This could be the case if every other model is insufficiently accurate for our context (Peruvian Spanish, etc.).

Case 2: Google Cloud Speech

Google Cloud Speech has a Python Interface called `google-cloud-speech`. It probably works the best out of all the options but it requires to have Google Application Credentials (that we can get free up to some extent) and an active internet connection. Its API is pretty intuitive and straightforward. It supports a variety of languages and dialects, including Spanish (Perú). More information at: `https://cloud.google.com/speech-to-text/docs/reference/libraries`.

Case 3: Kaldi (pykaldi)

Kaldi is a speech recognition toolkit which has Python wrappers for the C++ code in Kaldi and OpenFst libraries. It seems to be the preferred choice in the academic world. It has a Caribbean Spanish model with 30% WER (Word Error Rate), which is considerably higher than the best English model (5% WER). It should be enough, however.

It provides low-level access to the algorithms and their parameters, which means that, if done correctly, can increase the performance of our system. Depending on the requirements of ARCA, we might need to have such low-level control. However, it might not be the best choice to start with Kaldi, there are easier options that have more simple APIs.

Case 4: SpeechRecognition

Speech Recognition is a high-level library with support of several engines and APIs, online and offline:

- CMU Sphinx (works offline)
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection (works offline)

Combined with PyAudio and CMU Sphinx this library can be a great starting point. A very comprehensive tutorial can be found at 3.

An example:

```
import speech_recognition as sr

r = sr.Recognizer()
```

```
with sr.AudioFile('path/to/audiofile.wav') as source:
    audio = r.record(source)

r.recognize_google(audio, language='fr-FR')
```

Case 5: Sphinx (pocketsphinx)

CMU Sphinx also has its own simple Python wrapper called PocketSphinx which is really easy to use. A very practical tutorial can be found at 2. It is tremendously simple and has a really useful iterator class that allows live recognition through a microphone.

An example:

```
from pocketsphinx import LiveSpeech
for phrase in LiveSpeech():
    print(phrase)
```

Sources

1. Tensorflow, Simple Audio Recognition. Recovered at: `https://www.tensorflow.org/tutorials/sequen`
2. eonidi, Reconocimiento de voz (tiempo real) en idioma español con PocketSphinx. Recovered at: `http://blog.eonidi.com/index.php/2018/01/24/reconocimiento-de-voz-ti`
3. Real Python, The Ultimate Guide To Speech Recognition With Python. Recovered at: `https://realpython.com/python-speech-recognition/`