
scikit-surgeryspeech Documentation

Matt Clarkson

Mar 01, 2022

Contents

1	scikit-surgeryspeech	3
1.1	Example usage	4
1.2	Developing	4
1.3	Installing	5
1.4	Licensing and copyright	6
1.5	Acknowledgements	6
2	Voice Recognition Service	7
	Index	9

Source [code](#) is available on GitHub.



Author: Kim-Celine Kahl

scikit-surgeryspeech is part of the [SciKit-Surgery](#) software project, developed at the [Wellcome EPSRC Centre for Interventional and Surgical Sciences](#), part of [University College London \(UCL\)](#).

scikit-surgeryspeech supports Python 3.6.

scikit-surgeryspeech is a project which runs the [Python Speech Recognition API](#) in the background listening for a specific command. After saying the keyword you can say different commands, which get converted to QT Signals.

The speech recognition is done by the [Google Cloud API](#), you have to get the credentials to use it or change the recognition service.

Keyword detection is done by the [Porcupine API](#). This should be have been installed automatically via the `pvporcupine` dependency

Please explore the project structure, and implement your own functionality.

1.1 Example usage

To run an example, just start

```
sksurgeryspeech.py -c example_config.json
```

The config file should define the paths for the porcupine library and the Google Cloud API if you are using it.

You can then say the keyword depending on the Porcupine keyword file you chose and afterwards a command. The command “quit” exits the application.

Note: each time you have already entered a command, you need to say the keyword again to trigger the listening to commands.

1.2 Developing

1.2.1 Cloning

You can clone the repository using the following command:

```
git clone https://github.com/SciKit-Surgery/scikit-surgeryspeech
```

If you have problems running the application, you might need to install `portaudio`

Mac

```
brew install portaudio
```

Ubuntu

```
sudo apt-get install libasound-dev portaudio19-dev
```

If you’re going to try `sphinx` might need to install `pulseaudio-dev`

Ubuntu

```
sudo apt-get install swig libpulse-dev
```

1.2.2 Set up the Porcupine keyword detection

Then, you have to set the following variables in the configuration file

```
"porcupine dynamic library path" : ".tox/py37/lib/python3.7/site-packages/pvporcupine/  
→lib/linux/x86_64/libpv_porcupine.so",  
  "porcupine model file path" : ".tox/py37/lib/python3.7/site-packages/pvporcupine/  
→lib/common/porcupine_params.pv",  
  "porcupine keyword file" : [".tox/py37/lib/python3.7/site-packages/pvporcupine/  
→resources/keyword_files/linux/jarvis_linux.ppn"],
```


You can also [generate your own keyword files](#)

If you are using the speech recognition service within your own application, you have to start a background thread which calls the method to listen to the keyword over and over again.

You can find an example how to create such a thread in the `skurgeryspeech_demo.py`

1.2.3 Use the Google Cloud speech recognition service

To use the Google Cloud speech recognition service, you need to [get the credentials](#) first. After signing up, you should get a json file with your credentials. Download this file and add it to the configuration file

```
"google_credentials_file" : "snappy-speech-6ff24bf3e262.json",
```

To the path of your json file. You should then be able to run the application.

1.2.4 Change speech recognition service

You can try different speech recognition services by changing the recogniser entry in the config file. `sphinx`, `google` and `google_cloud` have all been tested, other options are possible but may not be implemented yet.

```
"recogniser" : "sphinx"  
"recogniser" : "google"  
"recogniser" : "google_cloud"  
"recogniser" : "wit"  
"recogniser" : "bing"  
"recogniser" : "azure"  
"recogniser" : "houndify"  
"recogniser" : "ibm"
```

1.2.5 Python development

This project uses tox. Start with a clean python environment, then do:

```
pip install tox  
tox
```

and the commands that are run can be found in `tox.ini`.

1.3 Installing

You can pip install directly from the repository as follows:

```
pip install git+https://github.com/SciKit-Surgery/scikit-surgeryspeech
```

1.3.1 Contributing

Please see the [contributing guidelines](#).

1.3.2 Useful links

- [Source code repository](#)

1.4 Licensing and copyright

Copyright 2019 University College London. scikit-surgeryspeech is released under the BSD-3 license. Please see the [license file](#) for details.

1.5 Acknowledgements

Supported by [Wellcome](#) and [EPSRC](#).

Voice Recognition Service

Speech API algorithm

class `sksurgeriespeech.algorithms.voice_recognition_service.VoiceRecognitionService` (*config: dict*)

Bases: `PySide2.QtCore.QObject`

Voice Recognition service which takes an microphone input and converts it to text by using the Google Cloud Speech-to-Text API.

Configuration dictionary must contain the following keys:

porcupine dynamic library path: `Porcupine/lib/<operating_system>/<processor_type>/<library_file>`

porcupine model file path: `Porcupine/lib/common/porcupine_params.pv`

porcupine keyword file(s): `Porcupine/resources/keyword_files/<operating_system>/<keyword>`

optional keys:

google credentials file: json file with google cloud api credentials

recogniser: api to use, options are sphinx, google, google_cloud, bing, houdify, ibm, wit

sphinx keywords: a list of keywords and sensitivities for sphinx timeout for command: default None

google_api_not_understand

google_api_request_failure

listen_for_keyword()

This method is called every 100 milliseconds by the QThread running and listens for the keyword

listen_to_command()

This method gets called when a specific command is said. It then listens for specific commands and converts them to QT Signals

request_stop()

Called by external client to stop timer.

```
run()  
    Entry point for the QThread which starts the timer to listen in the background  
start_listen  
start_processing_request  
staticMetaObject = <PySide2.QtCore.QMetaObject object>  
stop_timer  
voice_command
```

G

`google_api_not_understand` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 7

`google_api_request_failure` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 7

L

`listen_for_keyword()` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* method), 7

`listen_to_command()` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* method), 7

R

`request_stop()` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* method), 7

`run()` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* method), 7

S

`skurgeryspeech.algorithms.voice_recognition_service` (*module*), 7

`start_listen` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 8

`start_processing_request` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 8

`staticMetaObject` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 8

`stop_timer` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 8

V

`voice_command` (*skurgeryspeech.algorithms.voice_recognition_service.VoiceRecognitionService* attribute), 8