PocketSphinx Wrapper

- Overview
- Users
  - Using command line parameters
  - Creating a language model for PocketSphinx Wrapper
- Known Issues
- Message API
- FAQ

Overview

The pocketsphinx-sonic-server is a wrapper over the PocketSphinx speech recognition system which allows us to communicate with it using the Sonic protocol. PocketSphinx itself is a small-footprint continuous speech recognition system, suitable for handheld and desktop applications. We can use a client such as AcquireSpeech in order to communicate with pocketsphinx-sonic-server and get the text for the speech.

It is fairly easy to set up a language model for PocketSphinx and use it for speech recognition.

Quick facts:

- Location: /core/pocketsphinx-sonic-server/
- Language: C++
- Distribution: Binary
- Platform(s): *Windows*

Users

Using command line parameters

The applications expects a config file to be specified on the command line as follows

- -c [file-name]
  This config file should contain the following information specified as below
- -fwdflat
- -bestpath
- -lm [the language model file to be used]
- -dict [the dictionary to be used]
- -hmm [the acoustic mode]
  By default, the Virtual Human Toolkit uses the wall street journal acoustic model that comes with pocketsphinx and the CMU pronunciation dictionary. You can change this to use your own.
- -samprate [the sampling rate]

Creating a language model for PocketSphinx Wrapper

You will need to follow the below steps for creating your own language model for use with the PocketSphinx Wrapper.

- Go to the "data\pocketsphinx" folder within the Toolkit
- There is a file called corpus.txt which contains the utterances/lines to be spoken by the character. Replace this file with a file containing the new lines which you want the character to speak. Each utterance is one line in the file
- Now double click the batch file called generate_language_model.bat
- This will generate a new lm.arpa file which will be the new language model containing the lines in corpus.txt
- Now if you relaunch the PocketSphinx Wrapper from the launcher, it will reference this new language model that you just created

Known Issues

Message API

Sends:

- vrAllCall
- vrKillComponent

Receives:

- vrProcEnd

FAQ

See Main FAQ for frequently asked questions regarding the installer. Please use the Google Groups emailing list for unlisted questions.