

mod_pocketsphinx

About

[Pocketsphinx](#) is an open source speech recognition engine developed by Carnegie Mellon University. mod_pocketsphinx allows FreeSWITCH™ to recognize speech.

- Works on Windows, Mac OS X and Linux.
- 8k and 16k acoustical models.
- Semi-continuous recognition.
- Great for smaller grammars.

Install & Configure

1. Please update to at least rev 9194 so this will work correctly. Scoring was changed to be 0 = bad and 100 = good.
2. Build FreeSWITCH™ and enable mod_pocketsphinx
3. FreeSWITCH™ will automatically download and install pocketsphinx
4. enable mod_pocketsphinx in the [Modules.conf.xml](#)

Grammar Files

- Version 1.0.4 uses JSGF grammar files.
- More information about formatting can be found [here](#).

pizza_yesno.gram

```
#JSGF V1.0;

/**
 * JSGF Grammar for pizza_size
 */

grammar pizza_yesno;

<yes> = [ yes | yep | correct ];
<no> = [ no | nope ];

public <yesno> = <yes> <no>;
```

Setting up the Pizza Demo

- copy the demo scripts from the source to your working directory

```
cp -drp <freeswitch-src-dir>/scripts/javascript/js_modules /usr/local/freeswitch/scripts/
cp <freeswitch-src-dir>/scripts/javascript/ps_pizza.js /usr/local/freeswitch/scripts/
```

- if you are doing this on an old install you must copy the pocketsphinx.conf.xml to the conf directory

```
cp /usr/src/freeswitch/conf/autoload_configs/pocketsphinx.conf.xml /usr/local/freeswitch/conf/autoload_configs/
```

- Download the sounds files from [here](#)
- Move extracted pizza directory to sounds directory under freeswitch install (eg, /usr/local/freeswitch/sounds/en/us)
- Newer FreeSWITCH versions already contain /usr/local/freeswitch/conf/dialplan/default/00_pizza_demo.xml which sets up 74992 or "pizza" as an extension. If you are on an older FreeSWITCH version, make an extension like this:

```

<include>
<extension name="pizza_demo">
  <condition field="destination_number" expression="^(pizza|74992)$"/>
  <condition field="{module_exists(mod_spidermonkey)}" expression="true"/>
  <condition field="{module_exists(mod_pocketsphinx)}" expression="true">
    <action application="javascript" data="ps_pizza.js"/>
  </condition>
</extension>
</include>

```

- edit your ps_pizza.js with the location of your sound files

```
asr.setAudioBase("en/us/pizza/");
```

- Install grammar files

```

cd /usr/local/freeswitch/grammar
wget http://files.freeswitch.org/pizza_gram.tar.gz
tar xvzf pizza_gram.tar.gz

```

- Give it a try by calling extension 74992 and watching the console for messages.

Other info

Mod_pocketsphinx will build in the standard build on Linux and Mac. Yet to be tested on windows.

confidence score is 0+ higher numbers = more confidence.

```

2008-07-10 18:29:02 [DEBUG] switch_core_state_machine.c:140 switch_core_standard_on_execute() sofia/internal
/1000@10.0.1.110 Execute javascript(ps_pizza.js)
2008-07-10 18:29:02 [DEBUG] sofia_glue.c:1667 sofia_glue_activate_rtp() AUDIO RTP [sofia/internal/1000@10.
0.1.110] 10.0.1.110 port 21642 -> 10.0.1.17 port 57226 codec: 0 ms: 20
2008-07-10 18:29:02 [DEBUG] switch_rtp.c:741 switch_rtp_create() Starting timer [soft] 160 bytes per 20000ms
2008-07-10 18:29:02 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:02 [NOTICE] mod_spidermonkey.c:2014 session_answer() Channel [sofia/internal/1000@10.0.1.110]
has been answered
2008-07-10 18:29:02 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:02 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:02 [DEBUG] sofia.c:1845 sofia_handle_sip_i_state() Channel sofia/internal/1000@10.0.1.110
entering state [completed]
2008-07-10 18:29:02 [DEBUG] sofia.c:1845 sofia_handle_sip_i_state() Channel sofia/internal/1000@10.0.1.110
entering state [ready]
2008-07-10 18:29:04 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:05 [DEBUG] switch_core_media_bug.c:227 switch_core_media_bug_add() Attaching BUG to sofia
/internal/1000@10.0.1.110
2008-07-10 18:29:05 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:05 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:08 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:08 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:09 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1

```

```
channels 20ms
2008-07-10 18:29:10 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: TAKEOUT, Score: 44
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_order" score="44">
  <result name="match">TAKEOUT</result>
  <input>TAKEOUT</input>
</interpretation>
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [TAKEOUT]
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 44/1/75
2008-07-10 18:29:10 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [0] TAKEOUT =~ [Delivery:::
Delivery]
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [1] TAKEOUT =~ [Takeout:::Pickup]
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:364 console_log() ----Adding Pickup
2008-07-10 18:29:10 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [2] TAKEOUT =~ [Pickup:::Pickup]
2008-07-10 18:29:10 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:11 [DEBUG] SpeechTools.jm:109 console_log() Unloading grammar pizza_order
2008-07-10 18:29:12 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:16 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:16 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:19 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: EXTRA LARGE,
Score: 65
2008-07-10 18:29:19 [DEBUG] mod_pocketsphinx.c:327 pocketsphinx_asr_resume() Manually Resuming
2008-07-10 18:29:19 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_size" score="65">
  <result name="match">EXTRA LARGE</result>
  <input>EXTRA LARGE</input>
</interpretation>
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [EXTRA LARGE]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 65/1/75
2008-07-10 18:29:19 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [0] EXTRA LARGE =~
[^\s*Large:::ExtraLarge]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Adding ExtraLarge
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [1] EXTRA LARGE =~ [^\s*Large:::
Large]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [2] EXTRA LARGE =~ [^\s*Medium:::
Medium]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [3] EXTRA LARGE =~ [^\s*Small:::
Small]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [4] EXTRA LARGE =~ [^\s*Humongous:::
TotallyHumongous]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [5] EXTRA LARGE =~ [^\s*Huge:::
TotallyHumongous]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [6] EXTRA LARGE =~
[^\s*TotallyHumongous:::TotallyHumongous]
2008-07-10 18:29:19 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [7] EXTRA LARGE =~ [^\s*Totally:::
TotallyHumongous]
2008-07-10 18:29:19 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:20 [DEBUG] SpeechTools.jm:109 console_log() Unloading grammar pizza_size
2008-07-10 18:29:21 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:26 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:26 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:32 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: CHICAGO STYLE,
Score: 67
2008-07-10 18:29:32 [DEBUG] mod_pocketsphinx.c:327 pocketsphinx_asr_resume() Manually Resuming
2008-07-10 18:29:32 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_crust" score="67">
  <result name="match">CHICAGO STYLE</result>
  <input>CHICAGO STYLE</input>
</interpretation>
```

```
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [CHICAGO STYLE]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 67/1/75
2008-07-10 18:29:32 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [0] CHICAGO STYLE ==
[^Hand\s*Tossed$:::HandTossed]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [1] CHICAGO STYLE == [^Tossed$:::
HandTossed]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [2] CHICAGO STYLE ==
[^Chicago\s*style$:::Pan]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Adding Pan
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [3] CHICAGO STYLE == [^Chicago$:::
Pan]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [4] CHICAGO STYLE == [^Deep:::Pan]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [5] CHICAGO STYLE == [^Pan:::Pan]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [6] CHICAGO STYLE == [^Baked:::Pan]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [7] CHICAGO STYLE ==
[^New\s*York:::Thin]
2008-07-10 18:29:32 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [8] CHICAGO STYLE == [^Thin:::Thin]
2008-07-10 18:29:32 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:33 [DEBUG] SpeechTools.jm:109 console_log() Unloading grammar pizza_crust
2008-07-10 18:29:35 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:39 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:39 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:41 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: SPECIALTY PIZZA,
Score: 73
2008-07-10 18:29:41 [DEBUG] mod_pocketsphinx.c:327 pocketsphinx_asr_resume() Manually Resuming
2008-07-10 18:29:41 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_type" score="73">
  <result name="match">SPECIALTY PIZZA</result>
  <input>SPECIALTY PIZZA</input>
</interpretation>
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [SPECIALTY PIZZA]
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 73/1/75
2008-07-10 18:29:41 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [0] SPECIALTY PIZZA ==
[^Specialty$:::Specialty]
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [1] SPECIALTY PIZZA ==
[^Specialty\s*pizza$:::Specialty]
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:364 console_log() ----Adding Specialty
2008-07-10 18:29:41 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [2] SPECIALTY PIZZA == [^pick:::
Custom]
2008-07-10 18:29:41 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:42 [DEBUG] SpeechTools.jm:109 console_log() Unloading grammar pizza_type
2008-07-10 18:29:44 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:48 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:48 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:55 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: HAWAIIAN PIZZA,
Score: 78
2008-07-10 18:29:55 [DEBUG] mod_pocketsphinx.c:327 pocketsphinx_asr_resume() Manually Resuming
2008-07-10 18:29:55 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_specialty" score="78">
  <result name="match">HAWAIIAN PIZZA</result>
  <input>HAWAIIAN PIZZA</input>
</interpretation>
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [HAWAIIAN PIZZA]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 78/1/75
2008-07-10 18:29:55 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----We need to confirm this one
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [0] HAWAIIAN PIZZA == [^Hawaii:::
Hawaiian]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Adding Hawaiian
```

```
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [1] HAWAIIAN PIZZA =~ [^Hawaiian:::
Hawaiian]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [2] HAWAIIAN PIZZA =~ [^Meat:::
MeatLovers]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [3] HAWAIIAN PIZZA =~ [^Pickle:::
Pickle]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [4] HAWAIIAN PIZZA =~ [^World:::
Pickle]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [5] HAWAIIAN PIZZA =~ [^Salvador:::
Dali]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [6] HAWAIIAN PIZZA =~ [^Dolly:::
Dali]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [7] HAWAIIAN PIZZA =~ [^Dali:::
Dali]
2008-07-10 18:29:55 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [1] [8] HAWAIIAN PIZZA =~ [^Veg:::
Vegetarian]
2008-07-10 18:29:55 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:29:56 [DEBUG] SpeechTools.jm:109 console_log() Unloading grammar pizza_specialty
2008-07-10 18:29:58 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:01 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:01 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:01 [DEBUG] sofia.c:194 sofia_event_callback() event [nua_i_state] status [0][INVITE sent]
session: sofia/internal/1000@10.0.1.110
2008-07-10 18:30:01 [DEBUG] sofia.c:1845 sofia_handle_sip_i_state() Channel sofia/internal/1000@10.0.1.110
entering state [calling]
2008-07-10 18:30:01 [DEBUG] sofia.c:1845 sofia_handle_sip_i_state() Channel sofia/internal/1000@10.0.1.110
entering state [ready]
2008-07-10 18:30:03 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: YES, Score: 66
2008-07-10 18:30:03 [DEBUG] mod_pocketsphinx.c:327 pocketsphinx_asr_resume() Manually Resuming
2008-07-10 18:30:03 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_ynsno" score="66">
  <result name="match">YES</result>
  <input>YES</input>
</interpretation>
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [YES]
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 66/1/80
2008-07-10 18:30:03 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [0] YES =~ [^yes:::yes]
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:364 console_log() ----Adding yes
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [1] YES =~ [^correct:::yes]
2008-07-10 18:30:03 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [2] YES =~ [^no:::no]
2008-07-10 18:30:03 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:04 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:07 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:07 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:07 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:08 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:08 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:08 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:09 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:09 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:09 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:10 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:10 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1
channels 20ms
2008-07-10 18:30:10 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal
/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:11 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
```

```

2008-07-10 18:30:11 [DEBUG] SpeechTools.jm:109 console_log() Unloading grammar pizza_yesno
2008-07-10 18:30:11 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:12 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1 channels 20ms
2008-07-10 18:30:13 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:13 [DEBUG] switch_core_session.c:430 switch_core_session_receive_message() Kill sofia/internal/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:14 [DEBUG] mod_pocketsphinx.c:374 pocketsphinx_asr_get_results() Recognized: YES, Score: 57
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:150 console_log() ----XML:
<interpretation grammar="pizza_yesno" score="57">
  <result name="match">YES</result>
  <input>YES</input>
</interpretation>
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:150 console_log() ----Heard [YES]
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:150 console_log() ----Hit score 57/1/80
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [0] YES =~ [^yes:::yes]
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:364 console_log() ----Adding yes
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [1] YES =~ [^correct:::yes]
2008-07-10 18:30:14 [DEBUG] SpeechTools.jm:364 console_log() ----Testing [0] [2] YES =~ [^no:::no]
2008-07-10 18:30:15 [DEBUG] switch_ivr_play_say.c:911 switch_ivr_play_file() Codec Activated L16@8000hz 1 channels 20ms
2008-07-10 18:30:19 [DEBUG] switch_ivr_play_say.c:1175 switch_ivr_play_file() done playing file
2008-07-10 18:30:19 [WARNING] mod_pocketsphinx.c:212 pocketsphinx_asr_close() Port Closed.
2008-07-10 18:30:19 [DEBUG] switch_core_media_bug.c:312 switch_core_media_bug_close() Removing BUG from sofia/internal/1000@10.0.1.110
2008-07-10 18:30:19 [NOTICE] switch_core_state_machine.c:157 switch_core_standard_on_execute() Hangup sofia/internal/1000@10.0.1.110 [CS_EXECUTE] [NORMAL_CLEARING]
2008-07-10 18:30:19 [DEBUG] switch_channel.c:1361 switch_channel_perform_hangup() Kill sofia/internal/1000@10.0.1.110 [KILL]
2008-07-10 18:30:19 [DEBUG] switch_core_session.c:720 switch_core_session_signal_state_change() Kill sofia/internal/1000@10.0.1.110 [BREAK]
2008-07-10 18:30:19 [DEBUG] switch_core_state_machine.c:430 switch_core_session_run() (sofia/internal/1000@10.0.1.110) State EXECUTE going to sleep
2008-07-10 18:30:19 [DEBUG] switch_core_state_machine.c:365 switch_core_session_run() sofia/internal/1000@10.0.1.110 Running State Change CS_HANGUP
2008-07-10 18:30:19 [DEBUG] switch_core_state_machine.c:393 switch_core_session_run() (sofia/internal/1000@10.0.1.110) State HANGUP
2008-07-10 18:30:19 [DEBUG] mod_sofia.c:264 sofia_on_hangup() Channel sofia/internal/1000@10.0.1.110 hanging up, cause: NORMAL_CLEARING
2008-07-10 18:30:19 [DEBUG] mod_sofia.c:296 sofia_on_hangup() Sending BYE to sofia/internal/1000@10.0.1.110
2008-07-10 18:30:19 [DEBUG] switch_core_state_machine.c:46 switch_core_standard_on_hangup() Standard HANGUP sofia/internal/1000@10.0.1.110, cause: NORMAL_CLEARING
2008-07-10 18:30:19 [DEBUG] switch_core_state_machine.c:393 switch_core_session_run() (sofia/internal/1000@10.0.1.110) State HANGUP going to sleep
2008-07-10 18:30:19 [DEBUG] switch_core_session.c:784 switch_core_session_thread() Session 2 (sofia/internal/1000@10.0.1.110) Locked, Waiting on external entities
2008-07-10 18:30:19 [NOTICE] switch_core_session.c:802 switch_core_session_thread() Session 2 (sofia/internal/1000@10.0.1.110) Ended
2008-07-10 18:30:19 [NOTICE] switch_core_session.c:804 switch_core_session_thread() Close Channel sofia/internal/1000@10.0.1.110 [CS_HANGUP]

```

Acoustic Model for german language

An acoustic model describes a certain language on a phone base. A phone is something like a smallest distinguishable noise of a certain language. Dictionaries are used to sum up the phones to a word.

PocketSphinx comes with an english acoustic model which is to be used (of course) for the english language. For other languages you have to create your own acoustic model. This is a lot of work, especially creating the needed audio database (audio files, phone list, transcriptions and dictionary)

Voxforge (www.voxforge.org) offers, among other things, a german acoustic model under a GPL license found here: [\[1\]](#) Unfortunately it is not usable by PocketSphinx so we have to change it.

Based on Voxforge's audio data, the following lines describe how to build a PS compatible acoustic model (8kHz sample rate). It was tested on a CENTOS 5.3 x86_64 GNU/Linux system.

Requirements

Make sure the following is installed

- Python (e.g 2.4.3)
- flac (e.g. 1.1.2)

Download the following from voxforge.org

- [German audio files](#) (I used wget for this, but it's not really optimal, because it will download the whole web page ...)
- [German acoustic model configs](#)
- [CMU SphinxTrain 1.0](#)
- SphinxBase: This is delivered by freeswitch. But SphinxBase 0.4+ from [CMU](#) should work as well

Process

- Create work directory

```
mkdir <anywhere>/vf_de_test
cd <anywhere>/vf_de_test
```

- this new dir is now our <workdir>
- Prepare SphinxTrain

```
tar -jxf SphinxTrain-1.0.tar.bz2
cd SphinxTrain-1.0
make
cd ..
```

- Setup sphinx training environment "voxforge_de_sphinx"
- `./SphinxTrain-1.0/scripts_pl/setup_SphinxTrain.pl -task voxforge_de_sphinx`
- Content of <workdir>/

```
i drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 bin
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 bwaccumdir
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 etc
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 feat
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 logdir
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 model_architecture
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 model_parameters
drwxr-xr-x  3 ssw voip  4096  5. Aug 11:32 python
drwxr-xr-x 20 ssw voip  4096  5. Aug 11:32 scripts_pl
drwxr-xr-x  2 ssw voip  4096  5. Aug 11:32 wav
drwxr-xr-x 14 ssw voip  4096  5. Aug 11:02 SphinxTrain-1.0
-rw-r--r--  1 ssw voip 8297682 12. Feb 17:01 SphinxTrain-1.0.tar.bz2
```

- Copy Sphinxbase version from freeswitch source directory

```
cp -rf <path to FS source>/libs/sphinxbase-0.4.99 ./
cd sphinxbase-0.4.99
./autogen.sh
./configure --prefix=<workdir>/sphinxbase
make clean
make install
cd ..
```

- Extract acoustic model in a new directory

```
mkdir am_tmp
cd am_tmp
tar -zxvf AcousticModels.tgz
cd ..
Content of am_tmp:
```

```
-rw-r--r-- 1 ssw voip 7862417  8. Mai 12:27 AcousticModels.tgz
-rwxr-xr-x 1 ssw voip   3435 31. Mai 2008  espeak2phones.pl
drwxr-xr-x 2 ssw voip   4096  8. Mai 00:19  etc
drwxr-xr-x 3 ssw voip   4096  8. Mai 00:19  model_parameters
drwxr-xr-x 2 ssw voip   4096  8. Mai 00:19  result
drwxr-xr-x 2 ssw voip   4096  8. Mai 00:19  test
-rwxr-xr-x 1 ssw voip   1368 31. Mai 2008  traintest
```

- Preparing audio data (here 8kHz sample rate)
 - Put voxforge's audio archives to <workdir>/audio
 - Extract all archives

```
Cd audio
for i in *.tgz; do tar -zxvf $i; done
```

- Create script "copy_and_convert_audio.sh" in <workdir>


```

#Copyright 2009 Helmut Kuper
#
SOD=`pwd`
AD="${SOD}/audio"
TD="${SOD}/wav"

if [ ! -d $TD ]
then
    echo "ERROR: No wav directory found\n"
    echo "Please create it\n"
    exit 1
elif [ ! -d $AD ]
then
    echo "ERROR No audio directory found\n"
    exit 1
fi

copied=0
conv=0

cd $AD

for i in *
do
    if [ -d "$i/wav" ]
    then
        cd $i/wav
        for j in *.wav
        do
            cp $j "$TD/${i}_${j}"
            if [[ $(( copied++ % 100 )) -eq 0 ]]; then echo "wav: Copied: $((copied - 1))";
        fi
    done
    cd $AD
    elif [ -d "$i/flac" ]
    then
        cd $i/flac
        for j in *.flac
        do
            if [[ $j =~ '(.*)\.flac$' ]]
            then
                fname=${BASH_REMATCH[1]}
                #echo "Flac: Converting '$j' to ${i}_${fname}.wav"
                flac -f -s -d -o "$TD/${i}_${fname}.wav" $j
                if [[ $(( conv++ % 100 )) -eq 0 ]]; then echo "Flac: Converted $((conv
- 1))"; fi
            fi
        done
        cd $AD
    fi
done

cd $SOD

echo "Copied $copied files"
echo "Copied and converted $conv files"
echo "Copied $((copied + conv )) files to $TD"
echo
echo "Done"

```

- Converting (some are in flac format) and copy audio data to <workdir>/wav directory
- bash ./copy_and_convert_audio.sh (you must be in <workdir> directory)
- Create a feature file in <workdir>:
 - vi <workdir>/my_feat.params

```
-alpha 0.97
-dither yes
-doublebw no
-nfilt 40
-ncep 13
-lowerf 0
-upperf 4000
-nfft 512
-wlen 0.0256
-transform legacy
-feat s2_4x
```

- Create script for renaming MFC files in <workdir>.
 - vi <workdir>/renameMFC.sh

```
#Copyright 2009 Helmut Kuper
#
for i in *.chl.mfc
do
    if [[ $i =~ '(.*)\.chl\.mfc$' ]]
    then
        fname=${BASH_REMATCH[1]}
        mv $i $fname.mfc
        echo "Renaming '$i' to $fname.mfc"
    fi
done
echo "Done"
```

- Copy Voxforge's configurations to <workdir>/etc
 - cp /am_tmp/etc/* ./etc/
- Replace feature file with our own
 - cp ./my_feat.params ./etc/feat.params
- Adapt Voxforge's sphinx_train.cfg to our environment:
 - vi <workdir>/etc/sphinx_train.cfg

```
$CFG_BASE_DIR = "<workdir>/vf_de_test";
$CFG_SPHINXTRAIN_DIR = "./SphinxTrain-1.0";
#$CFG_HMM_TYPE = '.cont.'; # Sphinx III
$CFG_HMM_TYPE = '.semi.'; # Sphinx II
$CFG_LISTOFFILES = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.fileids";
$CFG_TRANSCRIPTFILE = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.transcription";
```

- Content of <workdir>

```
am_tmp
audio
bin
bwaccumdir
copy_and_convert_audio.sh
etc
feat
logdir
model_architecture
model_parameters
my_feat.params
python
renameMFC.sh
scripts_pl
sphinxbase
sphinxbase-0.4.99
SphinxTrain-1.0
SphinxTrain-1.0.tar.bz2
wav
```

- At least one File (openpentto-20080512-2_3_exp_5_1_Unit_0) is somehow corrupt, so delete line containing the name from:
 - ./etc/voxforge_de_sphinx_train.transcription
 - ./etc/voxforge_de_sphinx_train.fileids
 - Then delete the file "./wav/openpentto-20080512-2_3_exp_5_1_Unit_0.wav"
- Create MFC files of wav files
 - <workdir>/sphinxbase/bin/sphinx_fe `cat ./etc/feat.params` -c ./etc/voxforge_de_sphinx_train.fileids -di ./wav -do ./feat/ -ei wav -eo mfc -raw no -mswav yes -samprate 8000

INFO: cmd_ln.c(510): Parsing command line:

```
./sphinxbase/bin/sphinx_fe \  
-alpha 0.97 \  
-dither yes \  
-doublebw no \  
-nfilt 40 \  
-ncep 13 \  
-lowerf 0 \  
-upperf 4000 \  
-nfft 512 \  
-wlen 0.0256 \  
-transform legacy \  
-feat s2_4x \  
-c ./etc/voxforge_de_sphinx_train.fileids \  
-di ./wav \  
-do ./feat/ \  
-ei wav \  
-eo mfc \  
-raw no \  
-mswav yes \  
-samprate 8000
```

Current configuration:

[NAME]	[DEFLT]	[VALUE]
-alpha	0.97	9.700000e-01
-argfile		
-blocksize	200000	200000
-c		./etc/voxforge_de_sphinx_train.fileids
-cep2spec	no	no
-di		./wav
-dither	no	yes
-do		./feat/
-doublebw	no	no
-ei		wav
-eo		mfc
-example	no	no
-feat	sphinx	s2_4x
-frate	100	100
-help	no	no
-i		
-input_endian	little	little
-lifter	0	0
-logspec	no	no
-lowerf	133.33334	0.000000e+00
-mach_endian	little	little
-mswav	no	yes
-ncep	13	13
-nchans	1	1
-nfft	512	512
-nfilt	40	40
-nist	no	no
-nskip		
-o		
-raw	no	no
-remove_dc	no	no
-round_filters	yes	yes
-runlen		
-samprate	16000	8.000000e+03
-seed	-1	-1
-smoothspec	no	no
-spec2cep	no	no
-transform	legacy	legacy
-unit_area	yes	yes
-upperf	6855.4976	4.000000e+03
-verbose	no	no
-warp_params		
-warp_type	inverse_linear	inverse_linear
-whichchan	1	1
-wlen	0.025625	2.560000e-02



INFO: fe_interface.c(288): You are using the internal mechanism to generate the seed.

- Get rid of those ".ch1." parts in some MFC files
 - cd <workdir>/feat
 - bash ../renameMFC.sh
 - cd ..

You are now ready to start the training process. Before you do so, you can start a verification of all your provided data:

Execute „<workdir>/scripts_pl/00.verify/verify_all.pl“

```
MODULE: 00 verify training files
O.S. is case sensitive ("A" != "a").
Phones will be treated as case sensitive.
Phase 1: DICT - Checking to see if the dict and filler dict agrees with the phonelist file.
Found 3019 words using 41 phones
Phase 2: DICT - Checking to make sure there are not duplicate entries in the dictionary
Phase 3: CTL - Check general format; utterance length (must be positive); files exist
Phase 4: CTL - Checking number of lines in the transcript should match lines in control file
Phase 5: CTL - Determine amount of training data, see if n_tied_states seems reasonable.
Total Hours Training: 4.47290213675222
This is a small amount of data, no comment at this time
Phase 6: TRANSCRIPT - Checking that all the words in the transcript are in the dictionary
Words in dictionary: 3016
Words in filler dictionary: 3
Phase 7: TRANSCRIPT - Checking that all the phones in the transcript are in the phonelist, and all phones
in the phonelist appear at least once
```

Looks good so far. So let's start the training:

```

MODULE: 00 verify training files
O.S. is case sensitive ("A" != "a").
Phones will be treated as case sensitive.
  Phase 1: DICT - Checking to see if the dict and filler dict agrees with the phonelist file.
    Found 3019 words using 41 phones
  Phase 2: DICT - Checking to make sure there are not duplicate entries in the dictionary
  Phase 3: CTL - Check general format; utterance length (must be positive); files exist
  Phase 4: CTL - Checking number of lines in the transcript should match lines in control file
  Phase 5: CTL - Determine amount of training data, see if n_tied_states seems reasonable.
    Total Hours Training: 4.47290213675222
    This is a small amount of data, no comment at this time
  Phase 6: TRANSCRIPT - Checking that all the words in the transcript are in the dictionary
    Words in dictionary: 3016
    Words in filler dictionary: 3
  Phase 7: TRANSCRIPT - Checking that all the phones in the transcript are in the phonelist, and all phones
in the phonelist appear at least once
MODULE: 01 Vector Quantization
MODULE: 02 Training Context Independent models for forced alignment and VTLN
Skipped: $ST::CFG_FORCEDALIGN set to 'no' in sphinx_train.cfg
Skipped: $ST::CFG_VTLN set to '' in sphinx_train.cfg
MODULE: 03 Force-aligning transcripts
Skipped: $ST::CFG_FORCEDALIGN set to 'no' in sphinx_train.cfg
MODULE: 04 Force-aligning data for VTLN
Skipped: $ST::CFG_VTLN set to '' in sphinx_train.cfg
MODULE: 05 Train LDA transformation
Skipped (set $CFG_LDA_MLLT = 'yes' to enable)
MODULE: 06 Train MLLT transformation
Skipped (set $CFG_LDA_MLLT = 'yes' to enable)
MODULE: 20 Training Context Independent models
  Phase 1: Cleaning up directories:
    accumulator...logs...qmanager...models...
  Phase 2: Flat initialize
  Phase 3: Forward-Backward
    Baum welch starting for 1 Gaussian(s), iteration: 1 (1 of 1)
    0% 10% 20% 30% 40% 50% 60% 70% 80%

```

Now you can go and get a cup of coffee or tea or go to bed or...

[...]

For me the process ended with this:

```

[...]
Training for 1 Gaussian(s) completed after 4 iterations
MODULE: 90 deleted interpolation
  Phase 1: Cleaning up directories: logs...
  Phase 2: Doing interpolation...
WARNING: This step had 0 ERROR messages and 1 WARNING messages. Please check the log file for details.
  Phase 3: Dumping senones for PocketSphinx...
MODULE: 99 Convert to Sphinx2 format models
  Phase 1: Cleaning up old log files...
  Phase 2: Copy noise dictionary
  Phase 3: Make codebooks
0
  Phase 4: Make chmm files
  Phase 5: Make senone file
  Phase 6: Make phone and map files

```

The target folder "<workdir>/model_parameters/voxforge_de_sphinx.ci_semi" looks now like this:

```
feat.params
mdef
means
mixture_weights
noisedict
transition_matrices
variances
```

Then I copied those files to "<fs-folder>/grammar/model/de4/".

Further I copied "<workdir>/etc/voxforge_de_sphinx.dic" to "<fs-folder>/grammar/de4.dic" and created a grammar file which contained the words which should be recognized.

Finally I configured "pocketsphinx.conf.xml" like this:

```
<configuration name="pocketsphinx.conf" description="PocketSphinx ASR Configuration">
  <settings>
    <param name="threshold" value="400"/>
    <param name="silence-hits" value="25"/>
    <param name="listen-hits" value="1"/>
    <param name="auto-reload" value="true"/>
    <param name="narrowband-model" value="de4"/>
    <param name="wideband-model" value="wsj1"/>
    <param name="dictionary" value="de4.dic"/>
  </settings>
</configuration>
```

That's all you have to do as far as i know ... The results on my side were ... erm well ... suboptimal. After reloading mod_pocketsphinx FS detected simple german words but not very reliable. I think this is because of the small amount of prepared german audio data. Voxforge recommends 130 hours for training, but currently (March 2011) there are only 25hours available.

See Also

- [mod_dptools: play_and_detect_speech](#)