Sphinx Decoder Tutorial

March 10, 2010 § 7 Comments

I think the best way to understand how Sphinx3 is works is to follow the tutorial that was done by Keith Vertanen, Sphinx Simple Record. In this tutorial, a very basic recognizer is built that uses Sphinx to decode live speech and even plays back the recording at the end. With this code, it is easy to understand how to use the Sphinx APIs as it is a barebones set up of how Sphinx is run using the installed libraries.

The problem is that the code he has up will not work out of the box since the Sphinx team has changed some of the APIs since he did the project. I compiled the project for Ubuntu 9.10 (check my previous post for set up instructions) and here are the modifications that I needed to do to get it working.

Create a parent folder where all the subsequent code folders will be placed. For example, I made a folder called SphinxTutorial on my Desktop. Within Keith's project folders are files called install.linux. It has the instructions, that I will be rewriting with the modifications. Feel free to refer to them as there are some additional notes within it.

1. Download the program from http://keithv.com/software/simplerec/ to the SphinxTutorial folder.
2. Download Keith Vertanen's Port Audio Recorder to the SphinxTutorial folder from http://keithv.com/software/portaudio/
3. Download the latest snapshot of Port Audio from http://www.portaudio.com/download.html to the SphinxTutorial folder. Port Audio is an open source cross platform audio I/O library, Keith's program uses this library.
4. Now we compile the Port Audio Library with the following commands within Terminal.
    1. cd ~/Desktop/SphinxTutorial/portaudio
    2. ./configure –prefix=/usr
    3. make all
    4. sudo make install
    5. ls -l /usr/lib/*portaudio* *(This just lists the libraries that were installed, you should see libportaudio.a, libportaudio.so, libportaudio.so.2)*
5. Next we install the C++ binding library
    1. cd ~/Desktop/SphinxTutorial/portaudio/bindings/cpp/build/gnu
    2. chmod u+x configure
    3. ./configure –prefix=/usr
    4. make all
    5. sudo cp -f ../../lib/libportaudiocpp.* /usr/lib
    6. ls -l /usr/lib/*portaudio* *(Should now display libportaudiocpp.a, libportaudiocpp.so)*
6. All the dependencies are installed for the Port Audio Recorder, so we can actually build it. However, there are some issues that need to be fixed. If you try to compile the code, it will fail with an error "'atoi' was not declared in this scope". This is because the atoi function is from stdlib. Just go into the PortAudioRecPlay.cpp in ~/Desktop/SphinxTutorial/PortAudioRecPlay

folder and add the line #include “stdlib.h” near the rest of includes.

1. cd ~/Desktop/SphinxTutorial/PortAudioRecPlay/
2. make
3. ./PortAudioRecPlay *(This will run the code, pressing enter will start recording and pressing enter again will stop and playback)*

7. Now we install the Sphinx decoder as the SphinxSimpleRec program depends on it.
    1. cd ~/Desktop/SphinxTutorial/
    2. mkdir sphinx
    3. CMU_ROOT=~/Desktop/SphinxTutorial/sphinx; export CMU_ROOT (This is a shortcut way of referencing that path. It is also used by the makefile when compiling SphinxSimpleRec, therefore it needs to be exported)
8. Install Sphinxbase
    1. cd $CMU_ROOT
    2. svn co https://cmusphinx.svn.sourceforge.net:/svnroot/cmusphinx/trunk/sphinxbase
    3. cd sphinxbase
    4. ./autogen.sh
    5. ./configure –prefix=$CMU_ROOT *(This allows the binary and libraries to be installed at the CMU_ROOT location instead of /usr/lib. Before running the next three commands, look at the $CMU_ROOT folder. Once the commands are run, you will see new folders emerge.)*
    6. make
    7. make check
    8. make install
9. Install Sphinx3
    1. cd $CMU_ROOT
    2. svn co https://cmusphinx.svn.sourceforge.net:/svnroot/cmusphinx/trunk/sphinx3
    3. cd sphinx3
    4. ./autogen.sh
    5. ./autogen.sh *(Yes, twice….check the sphinx3 readme)*
    6. ./configure –prefix=$CMU_ROOT
    7. make
    8. make check *(7 out of 32 failed for me)*
    9. make install
10. Install lm3g2dmp *(There is a newer version, but I haven’t tried using it….it is a tool called sphinx_lm_convert)*
    1. Go to http://www.speech.cs.cmu.edu/sphinx/download/nightly/lm3g2dmp.nightly.tar.gz
    2. Extract tarball to the $CMU_ROOT folder.
    3. cd $CMU_ROOT/lm3g2dmp
    4. make
    5. cp lm3g2dmp $CMU_ROOT/bin *(Just copying the lm3g2dmp to the bin folder with the other executables from Sphinx)*
11. Download the Wall Street Journal acoustic model and 5K gigaword trigram to the SphinxSimpleRec folder
    1. http://www.keithv.com/software/sphinx/us/sphinx_wsj_all_cont_3no_8000_16.zip
    2. http://www.keithv.com/software/giga/lm_giga_5k_nvp_3gram.zip

3. Unzip both files into the SphinxSimpleRec folder

12. Create a binary version of the language model
    1. $CMU_ROOT/bin/lm3g2dmp lm_giga_5k_nvp_3gram/lm_giga_5k_nvp_3gram.arpa lm_giga_5k_nvp_3gram
13. Test the system using sphinx3_livepretend executable. This will pretend that it is doing a live decode except we pump an audio file into it. Now, there is a problem with one of the folder names that will cause a segmentation fault if the executable is run.
    1. Navigate to ~/Desktop/SphinxTutorial/SphinxSimpleRec/model_parameters
    2. Rename the folder "wsj_all_cont_3no_8000_16″ to "wsj_all_cont_3no_8000_16.cd"
    3. cd ~/Desktop/SphinxTutorial/SphinxSimpleRec/
    4. $CMU_ROOT/bin/sphinx3_livepretend salad.ctl . wsj_5k_rec.cfg *(You should get a recognition result)*
14. The final part of this is to actually build the SphinxSimpleRec program. However, there are a great deal of errors that need to be handled before. The next set of modifications is specifically for the SphinxSimpleRec.cpp file.
    1. #include "string.h" – line 24
    2. "cmd_ln_t *config = NULL;" – line 58
    3. *Change* if (cmd_ln_parse_file(S3_DECODE_ARG_DEFS, argv[1], TRUE)) *to* if ((config = cmd_ln_parse_file_r(config, S3_DECODE_ARG_DEFS, argv[1], TRUE)) == NULL)
    4. *Change* pSphinxFrontEnd = fe_init_auto(); *to* pSphinxFrontEnd = fe_init_auto_r(config);
    5. *Change* s3_decode_init(&decoder) *to* s3_decode_init(&decoder, config)
    6. Change fe_close(pSphinxFrontEnd); *to* fe_free(pSphinxFrontEnd)
15. Next we need to update the makefile. The make file is in the SphinxSimpleRec folder. Open it and make the following changes.
    1. *Change* LDFLAGS = -lportaudiocpp -lpthread -lportaudio -L$(CMU_ROOT)/lib -ls3decoder -lsphinxad -lsphinxfe -lsphinxfeat -lsphinxutil -Wl,–rpath -Wl,/$(CMU_ROOT)/lib *to* LDFLAGS = -lportaudiocpp -lpthread -lportaudio -L$(CMU_ROOT)/lib -ls3decoder -lsphinxad -lsphinxbase
16. We also need to update the LD_LIBRARY_PATH.
    1. EXPORT LD_LIBRARY_PATH=$CMU_ROOT/lib
17. We can now just type "make" in the SphinxSimpleRec directory to compile and to run the program, type –
    1. ./SphinxSimpleRec wsj_5k_rec.cfg

This is the end of this tutorial, well it is more of a supplement to Keith Vertanen's tutorial. I hope it was helpful. I also just separated the sphinx3_livedecode code and I can run it independently. It a bit less complicated than this one. Once I streamline the process, I will post up a tutorial on how to run it.