IIT BOMBAY

CYBORG

Speech Alignment Engine

Jigar Gada, Nikul Prajapati

4/3/2013

This is the README file for using the Cyborg Speech Alignment Engine.

Contents

Cyborg version 1.2

What's new

Main class

==========
```
edu.iitb.cyborg.aligner.TreeFunctions.Alignment
```

New in version 1.2

==================
1. Change in backtracing algorithm.
- Restriction of backtracing only from last state with SIL has been removed.
- Backtracing algorithm in version 1.2 can backtrace from any end state, not necessarily of "SIL", with high forward probability.

Required installations

Windows

=======
1. Java version 1.7.0 or above

Installation Guide:
http://www.java.com/en/download/help/windows_offline_download.xml

2. OpenFST version 1.3.1 or above

Download from here: https://code.google.com/p/openfstwin/downloads/detail?name=openfstwin-1.2.7-src-binaries.zip&can=2&q=

Installation Guide:

- OpenFst package for windows comes with compiled binaries. Download OpenFst binaries from above specified link and add these binaries to windows path variable.

Directory name of the compiled binaries: ./Release

- Make sure that OpenFst libraries have been added correctly to path variable by executing some of the commands.
  eg: fstconnect, fstcompile

If these above commands are not throwing any error or output on screen, you have successfully added binaries to path variable.

NOTE: If you are getting some dll missing error eg: msvcr100.dll, install latest version of "Microsoft Visual C++ Redistributable Package" from the link here.

Linux

=====

1. **Java version 1.7.0 or above**

Installation Guide:
http://www.java.com/en/download/help/linux_install.xml

To upgrade Java in Linux:
http://www.liberiangeek.net/2012/04/install-oracle-java-runtime-jre-7-in-ubuntu-12-04-precise-pangolin/

2. **OpenFST version 1.3.3 or above**

Download from here: http://www.openfst.org/twiki/bin/view/FST/FstDownload

Installation Guide:
The simplest way to compile this package is:

- `cd' to the directory containing the package's source code and type `./configure' to configure the package for your system.

  Running `configure' might take a while. While running, it print some messages telling which features it is checking for.

- Type `make' to compile the package. Optionally, type `make check' to run any self-tests that come with the package.

- Type `make install' to install the programs and any data files and documentation.

- You can remove the program binaries and object files from the source code directory by typing `make clean'.

- Often, you can also type `make uninstall' to remove the installed files again.

NOTE: If you installed openfst on non-standard path, or installed it to the default /usr/local/lib, but your variant of Linux doesn't support libraries there, you may need to set it up and/or ldconfig the directory explicitly

```
#export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/lib
#ldconfig -v -n /usr/local/lib
```

If all else fails, try installing openfst to wherever your blend of Linux installs all its libraries, such as /usr/lib.

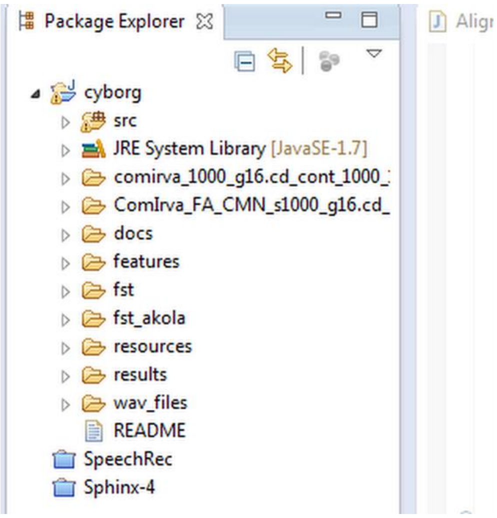
For more help refer: http://sourceforge.net/p/hfst/wiki/README/

Usage of CYBORG Aligner

Download Cyborg sphinx aligner from here: http://home.iitb.ac.in/~nicool/cyborg-1.1.jar

Source Code

To open Project in Eclipse
File -> Import -> General -> Existing Projects into the workspace -> Select directory *cyborg-1.2-src*
This will setup all the source files.



- cyborg
  - src
  - JRE System Library [JavaSE-1.7]
  - comirva_1000_g16.cd_cont_1000_
  - ComIrva_FA_CMN_s1000_g16.cd_
  - docs
  - features
  - fst
  - fst_akola
  - resources
  - results
  - wav_files
  - README
- SpeechRec
- Sphinx-4

Steps to run the code

Binary Sphinx Models

Folder: `edu.iitb.cyborg.preprocessing`
First Convert the Sphinx models into text file using *printp* command in Sphinx
http://www.speech.cs.cmu.edu/sphinxman/FAQ.html#22

Convert these text files into CYBORG compatible binary files by running the file ***StringtoBinary***.
I have provided some sample Sphinx text model files. These models are trained on limited native Indian data set. If you have you own Sphinx model, you can use their text format.

Arguments list definition for binary models:

[NAME]	[DEFLT]	[DESCR]
-mean	mean_file	
-variance	variance_file	
-tmat	transition_matrix file	
-mixwt	mixture_weights file	
-mdef	mdef file	

Provide the arguments by clicking on
Run -> Run configuration -> Arguments

E.g. Arguments can be entered as follows:

```
-mdef models/mdef -mean models/means.txt -variance
models/variances.txt -mixwt models/mixture_weights.txt -tmat
models/transition_matrices.txt
```

The final model files to be used for alignment will be:

- mean_bin
- variance_bin
- mixWt_bin
- tmat_bin
- mdef_tab

Feature Extraction

Folder: `edu.iitb.frontend.audio.feature`

You need the enter the parameters in the file *FeatureFileExtractor.java*
Default parameters:

```
sampleRate = 8000;
windowSize = 256;
// actual window size. Calculate it according to the sampling rate.
// by default it is 25.625ms which is equivalent to 205 samples for
// 8000 Hz sampling rate.
newWindowSize = 205;
// hopsize by default is 10ms
hopsize = 80;
//No of mel filters with the first coefficient as energy.
numberCoefficients = 13;
//Lower edge of filter
minFreq = 133;
//Upper edge of filter
maxFreq = 3500;
//Number of mel filter banks
numberFilters = 31;
```

Alignment

Folder: `edu.iitb.cyborg.aligner.TreeFunctions`

Arguments list definition for alignment:

[NAME]	[DEFLT]	[DESCR]
-models		Input model files directory
-ctl		Control file listing utterances to be processed
-t		Input transcriptions file corresponds to audio
-dict		Main pronunciation dictionary (lexicon) input file
-in	.	Input audio wav file(s) directory
-cepdir	.	Input cepstrum files directory

-triphn	yes	Determines whether to use triphone search or monophone search
-phseg	.	Output directory for phone segmentation files
-wdseg	.	Output directory for word segmentation files

Helper arguments

[NAME]	[DESCR]
-version	Shows the current version.
-help	Shows the help on usage.

e.g.

```
-models models/ -ctl resources\fileIds.txt -t
resources\transcription.txt -dict resources\Dictionary.dic -in
wav_files -cepdir features -triphn yes
```

Models -> folder which contains the binary models
fileIds.txt -> contains the name of the wav files (without the .wav extension)
transcription.txt -> contains the transcription of the wav file
Dictionary.dic -> Mapping of words to phonemes
in -> directory where the wav files defined in fileIds are stored
cepdir ->folder in which the extracted features will be stored

Go through these files carefully to understand it properly.

Future Tasks

- Reading files directly from Sphinx Binaries.
- Adding compatibility for HTK models.
- More documentation