Seminar: Voicebuilding for TTS Synthesis

Ingmar Steiner

WS 2013/14

Formalities

- This course is a **project seminar** (for LST/CoLi students), or a regular **seminar** (for CS/VC students).
- *Successful* participation of the lecture "Text-to-Speech Synthesis" (Prof. Möbius) is a **mandatory prerequisite**.
- To pass *this* course, you will need to build TTS voices and **submit them, along with a written report**. The report must explain the entire process, including any problems encountered, and their resolution (5 to 10 pp.). This report is due *two weeks* after the end of the seminar (**11.04.2014**).
- Register through LSF/HISPOS by **24.03.2014**.
- Mailing list for questions, discussion: `voicebuildingsem@ml.coli.uni-saarland.de`

Course Overview

- Split into 4 to 5 groups
- Design prompt list
- Record speech corpus in studio
- Process recordings (including automatic phonetic segmentation)
- Build TTS voices (unit-selection and HMM-based variants)
- Use MaryTTS[1] (*invented here*)

[1] https://github.com/marytts/marytts

Next: MaryTTS installation

MaryTTS

- Open-source, multilingual TTS platform implemented in Java
- `http://mary.dfki.de/` — outdated, to be replaced by...
- ...what is currently at `http://diax.coli.uni-saarland.de/mary/`
- Development hosted at `https://github.com/marytts/marytts`

Download MaryTTS

```
$ git clone git@github.com:psibre/marytts.git -b v5.1
beta2
Cloning into 'marytts'...
remote: Reusing existing pack: 43338, done.
remote: Counting objects: 2306, done.
remote: Compressing objects: 100% (789/789), done.
remote: Total 45644 (delta 1074), reused 2155 (delta
997)
Receiving objects: 100% (45644/45644), 133.55 MiB |
4.83 MiB/s, done.
Resolving deltas: 100% (30976/30976), done.
Checking connectivity... done.
Note: checking out '4
c476d380217ca7f667767e666f6231b39d820ca'.
$ cd marytts/
```

Install MaryTTS

```
$ mvn install
(lots of output)
[INFO] ------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------
[INFO] Total time: 1:21.645s
[INFO] Finished at: Sat Mar 15 18:34:34 CET 2014
[INFO] Final Memory: 93M/1638M
[INFO] ------------------------------------------------------------
```

Start MaryTTS server instance

```
$ cd target/marytts-5.1beta2/
$ export MARYBASE=$PWD
$ bin/marytts-server.sh
java version "1.7.0_40"
Java(TM) SE Runtime Environment (build 1.7.0_40-b43)
Java HotSpot(TM) 64-Bit Server VM (build 24.0-b56,
mixed mode)

MARY server 5.1-beta2 starting as a HTTP server...
started in 3.472 s
```

Verify by browsing to `http://localhost:59125/`.

Next: Voicebuilding Done Quick

Prepare to get the data

```
$ git clone git@bitbucket.org:psibre/cmu-slt-arctic-
data.git -b seminar
Cloning into 'cmu-slt-arctic-data'...
remote: Counting objects: 36, done.
remote: Compressing objects: 100% (34/34), done.
remote: Total 36 (delta 15), reused 0 (delta 0)
Receiving objects: 100% (36/36), 51.82 KiB | 0 bytes/s
, done.
Resolving deltas: 100% (15/15), done.
Checking connectivity... done.
$ cd cmu-slt-arctic-data/
```

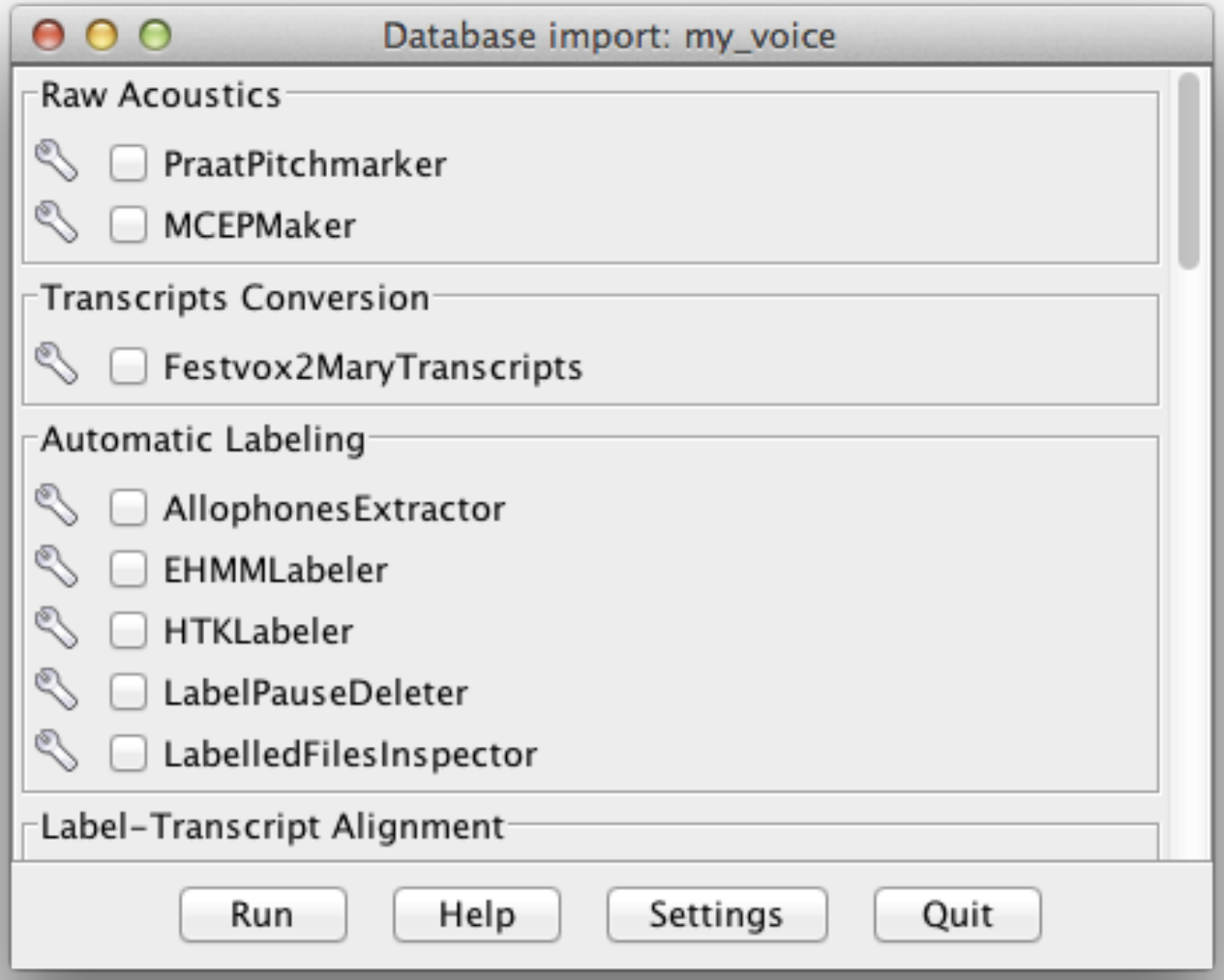
Download and unpack the data

```
$ ./gradlew
:downloadData
Download http://www.speech.cs.cmu.edu/cmu_arctic/
packed/cmu_us_slt_arctic-0.95-release.tar.bz2
:unpackData
:generateTxt

BUILD SUCCESSFUL

Total time: 1 mins 18.328 secs
```

Prepare voicebuilding directory

```
$ cd ..
$ mkdir voice-cmu-slt-arctic
$ cd voice-cmu-slt-arctic/
$ export VOICEDIR=$PWD
$ for dirname in wav text lab pm mcep
  do
    ln -sfv ../cmu-slt-arctic-data/$dirname
  done
./wav -> ../cmu-slt-arctic/wav
./text -> ../cmu-slt-arctic/text
./lab -> ../cmu-slt-arctic/lab
./pm -> ../cmu-slt-arctic/pm
./mcep -> ../cmu-slt-arctic/mcep
$ mv -v lab lab_raw
lab -> lab_raw
$ open https://gist.github.com/psibre/
abf0d2ac833046af17cb
$ mkdir lab
$ ./convert_labels.pl
```

Initialize voicebuilding



Run voicebuilding components (1/2)

1. FeatureSelection
2. AllophonesExtractor
3. PhoneUnitComputer
4. HalfPhoneUnitComputer
5. TranscriptionAligner
6. PhonUnitFeatureComputer
7. HalfPhonUnitFeatureComputer
8. PhoneLabelFeatureAligner
9. HalfPhoneLabelFeatureAligner
10. WaveTimelineMaker
11. BasenameTimelineMaker
12. MCepTimelineMaker
13. PhoneUnitfileWriter
14. PhoneFeatureFileWriter

Install Edinburgh Speech Tools

```
$ cd ..
$ wget http://www.cstr.ed.ac.uk/downloads/festival
/2.1/speech_tools-2.1-release.tar.gz
$ tar xfvz speech_tools-2.1-release.tar.gz
$ cd speech_tools/
$ ./configure
```
(autoconf output)
```
$ make
```
(compile, compile, ...)
```
$ export ESTDIR=$PWD
```

Run voicebuilding components (2/2)

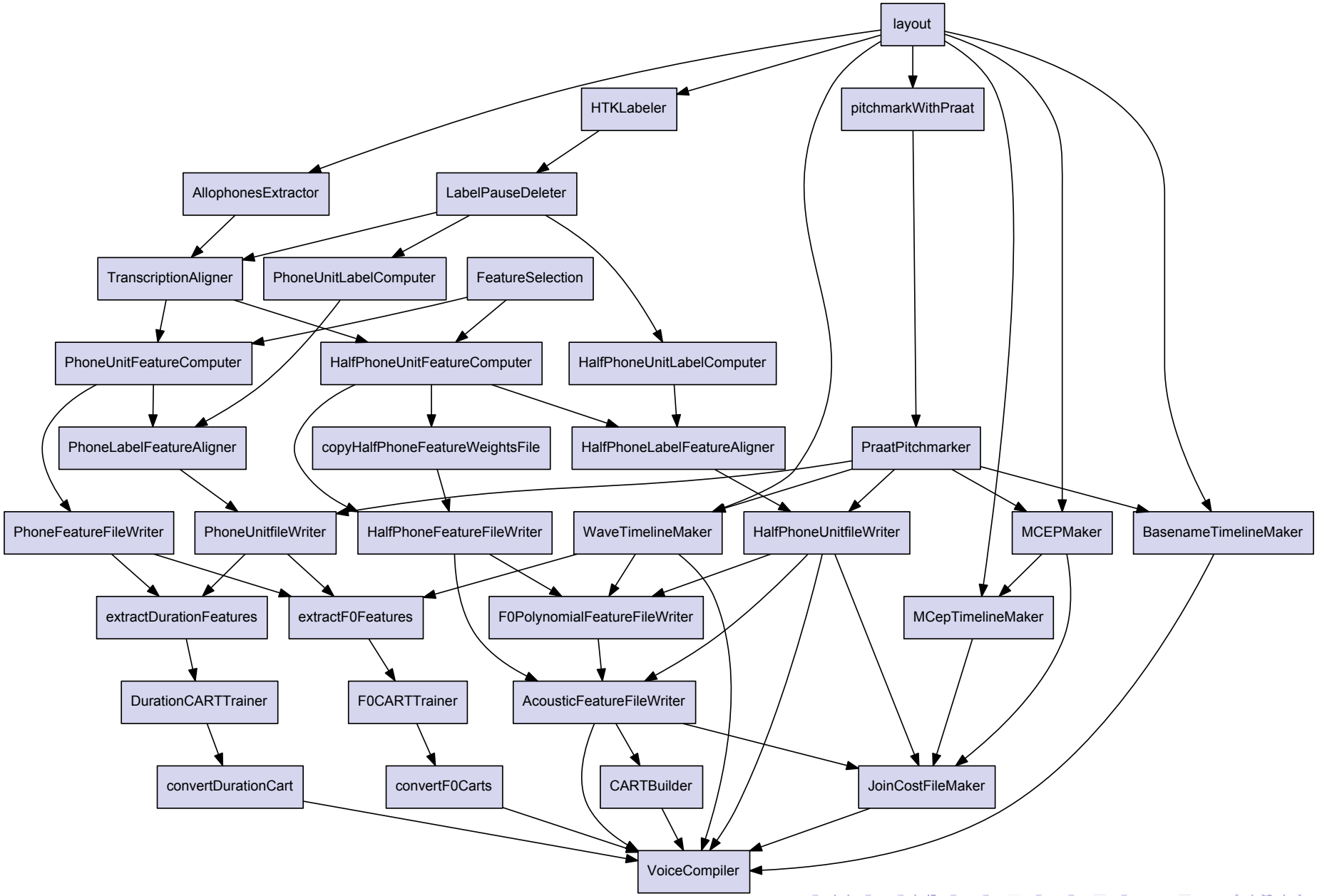
0. Configure `estDir=$ESTDIR`
1. DurationCARTTrainer
2. F0CARTTrainer
3. HalfPhoneUnitfileWriter
4. HalfPhoneFeatureFileWriter
5. F0PolynomialFeatureFileWriter
6. AcousticFeatureFileWriter
7. JoinCostFileMaker
8. CARTBuilder
9. VoiceCompiler

Install new voice

```
$ cd mary/voice-my_voice/target/
$ cp voice-my_voice-5.1-beta2-component.xml voice-
my_voice-5.1-beta2.zip $MARYBASE/download/
$ cd $MARYBASE
$ bin/marytts-component-installer.sh
my_voice selected for installation
Check license(s)
Downloading license from http://mary.dfki.de/download/
by-nd-3.0.html
Lookup took 153 ms
Showing license http://mary.dfki.de/download/by-nd
-3.0.html for 1 components
License accepted.
Starting installation
Now installing my_voice...
(unzipping output)
...done
```

Next: Instant Replay

Voicebuilding Task Execution Graph



Raw acoustics

1. Pitchmarking (using Praat)

   input `wav/*.wav`

   output `pm/*.pm`

2. MCEP coefficient extraction (using EST)

   input `wav/*.wav`

   output `mcep/*.mcep`

G2P and labeling

1. Predict phone sequence from text (using MaryTTS)
   - input `text/*.txt`
   - output `prompt_allophones/*.xml`
2. Phone-level segmentation
   - input `text/*.txt`, `wav/*.wav`
   - output `lab/*.lab`
3. Check alignment
   - input `prompt_allophones/*.xml`, `lab/*.lab`
   - output `allophones/*.xml`

Unit features

1. Select feature set

   output `mary/features.txt`

2. Compute and assign feature vector to each unit (using MaryTTS)

   input `allophones/*.xml`, `mary/features.txt`

   output `phonefeatures/*.pfeats`, `halfphonefeatures/*.hpfeats`

Data files

Compile "timeline" files for

- Audio samples

  input `wav/*.wav, pm/*.pm`
  output `mary/timeline_waveforms.mry`

- Utterances

  input `wav/*.wav, pm/*.pm`
  output `mary/timeline_basenames.mry`

- MCeps

  input `wav/*.wav, mcep/*.mcep`
  output `mary/timeline_mcep.mry`

These contain the actual data from the `wav` and `mcep` files, in pitch-synchronous "datagram" packets.

Acoustic models

- Phone-level unit file

  input `pm/*.pm`, `phonelab/*.lab`
  output `mary/phoneUnits.mry`

- Phone-level feature file

  input `phonefeatures/*.pfeats`,
  output `mary/phoneFeatures.mry`,
  `mary/phoneUnitFeatureDefinition.txt`

- CARTs for duration and F0

  input `mary/phoneUnits.mry`,
  `mary/phoneFeatures.mry`,
  `mary/timeline_waveforms.mry`
  output `mary/dur.tree`, `mary/f0.left.tree`,
  `mary/f0.mid.tree`, `mary/f0.right.tree`

Unit selection files (1/3)

- Halfphone-level unit file

  input `pm/*.pm, halfphonelab/*.hplab`
  output `mary/halfphoneUnits.mry`

- Halfphone-level feature file

  input `halfphonefeatures/*.hpfeats,`
  output `mary/halfphoneFeatures.mry,`
  `mary/halfphoneUnitFeatureDefinition.txt`

Unit selection files (2/3)

- F0 contour file

```
input  mary/halfphoneUnits.mry,
       mary/timeline_waveforms.mry,
       mary/halfphoneFeatures.mry
output mary/syllableF0Polynomials.mry
```

- Acoustic feature file

```
input  mary/halfphoneUnits.mry,
       mary/syllableF0Polynomials.mry,
       mary/halfphoneFeatures.mry
output mary/halfphoneFeatures_ac.mry,
       mary/halfphoneUnitFeatureDefinition_ac.txt
```

Unit selection files (3/3)

- Join cost file

  input `mcep/*.mcep`, `mary/timeline_mcep.mry`, `mary/halfphoneUnits.mry`, `mary/halfphoneFeatures_ac.mry`

  output `mary/joinCostFeatures.mry`, `mary/joinCostWeights.txt`

- Top-level CART

  input `mary/halfphoneFeatures_ac.mry`

  output `mary/cart.mry`, `featureSequence.txt`

Distributable voice package

Collect, filter resources, generate descriptor using Maven

input `mary/cart.mry`, `featureSequence.txt`, `mary/dur.tree`, `mary/f0.left.tree`, `mary/f0.mid.tree`, `mary/f0.right.tree`, `mary/halfphoneFeatures_ac.mry`, `mary/joinCostFeatures.mry`, `mary/joinCostWeights.txt`, `mary/halfphoneUnits.mry`, `mary/timeline_basenames.mry`, `mary/timeline_waveforms.mry`

output `my_voice.zip`, `my_voice-component.xml`

Next: your turn

Grouping

- Work in small groups 3 to 4 people.
- Each group should have at least
  - one native English speaker, and
  - one programmer/hacker, and
  - one phonetician

Speech recording

- Each group plans and carries out recordings for $\sim 1\,\mathrm{h}$ of speech data
- Use a phonetically balanced prompt set, e.g., TIMIT or ARCTIC
- Use collaborative versioning tools to share this data in the team, e.g., Dropbox, `git-annex`, etc.

Phonetic segmentation

Use forced alignment for automatic segmentation

- EHMM,
- HTK,
- MAUS,
- CMU Sphinx,
- Julius,
- Kaldi,
- ...

and let's not forget: *manual labor!*

Software dependencies

MaryTTS

- Java JDK (preferably 6 or 7)
- Maven 3
- GitHub

Acoustic analysis

- Praat (or WaveSurfer or ESPS)
- Edinburgh Speech Tools
- SoX

HMM-based voicebuilding

- HTK (with HDecode and HTS patch)
- HTS_engine
- SPTK
- Tcl (with SNACK library!)

At least some of this must be built from source, so GCC 4.5 (or so) is a must

Have fun!