Creating New Language and Voice Components for the Updated MaryTTS Text-to-Speech Synthesis Platform

Ingmar Steiner\textsuperscript{1-3}, Sébastien Le Maguer\textsuperscript{1,2}

\textsuperscript{1}Multimodal Computing and Interaction, Saarland University  \quad \textsuperscript{2}Computational Linguistics & Phonetics, Saarland University  \quad \textsuperscript{3}DFKI Language Technology Lab, Saarbrücken

1 Introduction

- MaryTTS is a widely used, open-source text-to-speech synthesis (TTS) system
- A redesigned architecture has been implemented
- New component building process (languages and voices)
- Processing leverages build automation (Gradle)

2 Plugin-based voice building (VB)

1. Data preparation/forced alignment (Kaldi)
2. Acoustic feature extraction (Praat/SPTK/EST)
3. Linguistic feature extraction (MaryTTS)
4. Model building
   - Unit selection (acoustic models, data packaging)
   - Statistical parametric speech synthesis (HMM)

3 Example

LEXICON RESOURCES

speech corpus  Bintray  Gradle Plugins Portal

obtains lexicon resource
applies lexicon-compiler-plugin
trains lexicon FST and G2P rules
publishes marytts-lexicon-xy

depends on marytts-lexicon-xy
provides NLP modules for language “xy”
publishes marytts-lang-xy

depends on marytts-lang-xy and speaker-somename-xy-data

depends on marytts-lang-xy and speaker-somename-xy-data

obtains audio and text resources
obtains phonetic annotation or applies kaldi-mfa-plugin
publishes speaker-somename-xy-data

1. extracts acoustic features from audio
2. extracts linguistic features from text
3. aligns features based on phonetic labels
4. builds models
5. packages data (for unit selection)
publishes voice-somename-xy

4 New configuration mechanism

default  given in the MaryTTS module
voice  configured in voicebuilding project
user  specified at runtime (takes precedence)

⇒ More flexible control over module pipeline, parameters, models assets, etc.

5 Conclusion

- New language and voice component building workflow
- Heavy use of state-of-the-art build automation
  + more efficient
  + more extensible (plugins)
- Entire process redesigned and extended (cloud-based)