Speech Synthesis: Overview

11-752: Class Overview
Overview

- Speech Phonetics Prosody Perception and Synthesis:
- Sub-topics:
  - Text analysis and normalization (Alan)
  - Lexicons and pronunciation modeling
  - Spectrogram reading (Maxine) Feb/Mar
  - Crowdsourcing in speech (plus project) (Maxine) Feb/Mar
  - Prosody Generation (Alan)
  - Waveform Generation (Alan)
  - Multilinguality and Limited Resources (Alan)
  - Evaluation (Alan)
  - Voice/Language/Style/Emotion conversion (Alan)
  - Project (Alan) May
Coursework

- Practical course: i.e. you **do** things
- Weekly courseworks:
  - Short exercises related to current topic
  - Dur Monday's at noon.
- Crowdsourcing Project (March)
- Final Project (May)
- Grading
  - Best 5 weekly courseworks (50%)
  - Crowdsourcing Project (20%)
  - Final Project (30%)
Examples

- Example weekly coursework
  - From given data build a duration model
  - Design a diphone prompt list for some language

- Crowdsourcing project
  - Challenge defined by Maxine

- Final project
  - Something novel in synthesis
  - Evaluation technique, machine learning, new language,
  - Singing, ...
First task

- Technically not a weekly coursework
  - But necessary for all other coursework
- Install the software and test it
  - Install SPTK and Edinburgh Speech Tools
  - Install Festival and FestVox voice building tools
- Any platform: Windows, OSX or Linux/Unix but
  - You'll want to hear the result (i.e. working audio)
  - So use a laptop/desktop rather than a server
- See instructions on course website (linked from awb's homepage)
- Ask if you get stuck
- And, test it by ....
Build a Talking Clock

- Build a talking synthesizer clock from your own voice
- Record 24 standard prompts
  - “The time is now, about five past one, in the morning”
  - “The time is now, just after ten past two, in the morning”
  - ...
- Use the FestVox limited domain tools to build a talking clock
- It will work with your accent and sound just like you
- You may have to use audacity to record your prompts
  - Make sure you generate 16KHz, Mono, .wav (RIFF) format
  - And name the files properly (time_0001.wav, time_0002.wav ...)
- Due Monday 20th Jan at noon by email to awb@cs.cmu.edu
- See detailed instructions on website.
- Ask if you get stuck
Physical Models

• Blowing air through tubes…
  – von Kemplen’s synthesizer 1791

• Synthesis by physical models
  – Homer Dudley’s Voder. 1939
More Computation – More Data

- **Formant synthesis (60s-80s)**
  - Waveform construction from components

- **Diphone synthesis (80s-90s)**
  - Waveform by concatenation of small number of instances of speech

- **Unit selection (90s-00s)**
  - Waveform by concatenation of very large number of instances of speech

- **Statistical Parametric Synthesis (00s-..)**
  - Waveform construction from parametric models
Waveform Generation

- Formant synthesis
- Random word/phrase concatenation
- Phone concatenation
- Diphone concatenation
- Sub-word unit selection
- Cluster based unit selection
- Statistical Parametric Synthesis
Building a Research Field

- **Tools**
  - Allow others to easily join the field

- **Common Data Sets**
  - Be able to concentrate on techniques
  - Have common comparisons

- **Evaluation**
  - Realistically compare techniques

- **Have Users**
  - Some one has to care about your results

- **Don’t become stifled**
  - Ensure there are new tasks and directions
Festival Speech Synthesis System

http://festvox.org/festival
General system for multi-lingual TTS
C/C++ code with Scheme scripting language
General replaceable modules
  lexicons, LTS, duration, intonation, phrasing,
  POS tagging tokenizing, diphone/unit selection
General Tools
  intonation analysis (F0, Tilt), signal processing
  CART building, n-grams, SCFG, WFST, OLS
No fixed theories
New languages without new C++ code
Multiplatform (Unix, Windows, OSX)
Full sources in distribution
Free Software
http://festvox.org

“I want it to speak like me!”

-Festival is an engine, how do you make voices
- Building Synthetic Voices
  - Tools, scripts, documentation
  - Discussion and examples for building voices
  - Example voice databases
  - Step by Step walkthroughs of processes
- Support for English and other languages
- Support for different waveform techniques:
  - diphone, unit selection, limit domain, HMM
- Other support: lexicon, prosody, text analysers
http://cmuflite.org

“But I want it to run on my phone!”
- FLITE a fast, small, portable run-time synthesizer
- C based (no loaded files)
- Basic FestVox voices compiled into C/data
- Thread safe
- Suitable for embedded devices
  - Ipaq, Linux, WinCE, PalmOS, Symbian
- Scalable:
  - quality/size/speed trade offs
  - frequency based lexicon pruning
- Sizes:
  - 2.4Meg footprint (code+data+runtime RAM)
  - < 0.025 secs “time-to-speak”
Common Data Sets

- **Data drive techniques need data**
- **Diphone Databases**
  - CSTR and CMU US English Diphone sets (kal and ked)
- **CMU ARCTIC Databases**
  - 1200 phonetically balanced utterances (about 1 hour)
  - 7 different speakers (2 male 2 female 3 accented)
  - EGG, phonetically labeled
  - Utterances chosen from out-of-copyright text
  - Easy to say
  - Freely distributable
  - Tools to build your own in your own language
Blizzard Challenge

- **Realistic evaluation**
  - Under the same conditions

- **Blizzard Challenge [Black and Tokuda]**
  - Participants build voice from common dataset
  - Synthesis test sentences
  - Large set of listening experiments
  - Since 2005, now in 9th year
  - 15-20 groups (Academia, Research Labs and Commercial Companies)
How to test synthesis

- **Blizzard tests:**
  - Do you like it? (MOS scores)
  - Can you understand it?
    - SUS sentence
    - The unsure steaks overcame the zippy rudder

- **Can’t this be done automatically?**
  - Not yet (at least not reliably enough)
  - But we now have lots of data for training techniques

- **Why does it still sound like robot?**
  - Need better (appropriate testing)
Speech Synthesis Techniques

- Unit selection
- Statistical parameter synthesis
- Automated voice building
  - Database design
  - Language portability
- Voice conversion
• Target cost and Join cost [Hunt and Black 96]
  – Target cost is distance from desired unit to actual unit in the databases
    • Based on phonetic, prosodic metrical context
  – Join cost is how well the selected units join
Cluster units [Donovan et al 96, Black et al 97]

\[
Adist(U, V) = \begin{cases} 
    Adist(V, U) & \text{if } |V| > |U| \\
    \frac{WD*|U|}{|V|} * \sum_{i=1}^{|U|} \sum_{j=1}^{n} \frac{W_j \cdot (abs(F_{ij}(U) - F_{i*|V|/|U|,j}(V)))}{SD_j * n * |U|}
\end{cases}
\]

- \(|U| = \text{number of frames in } U\)
- \(F_{xy}(U) = \text{parameter } y \text{ of frame } x \text{ of unit } U\)
- \(SD_j = \text{standard deviation of parameter } j\)
- \(W_j = \text{weight for parameter } j\)
- \(WD = \text{duration penalty}\)
• Cost metrics
  – Finding best weights, best techniques etc

• Database design
  – Best database coverage

• Automatic labeling accuracy
  – Finding errors/confidence

• Limited domain:
  – Target the databases to a particular application
  – Talking clocks
  – Targeted domain synthesis
Unit Selection
The “standard” method
“Select appropriate sub-word units from large databases of natural speech”

Parametric Synthesis: [NITECH: Tokuda et al]
HMM-generation based synthesis
Cluster units to form models
Generate from the models
“Take ‘average’ of units”
Unit Selection:

large carefully labelled database
quality good when good examples available
quality will sometimes be bad
no control of prosody

Parametric Synthesis:

smaller less carefully labelled database
quality consistent
resynthesis requires vocoder, (buzzy)
can (must) control prosody
model size much smaller than Unit DB
Parametric Synthesis

- Probabilistic Models

\[ \text{argmax}(P(O|W)) \]

- Simplification

\[ \text{argmax}(P(o_0|W), P(o_1|W), ..., P(o_n|W)) \]

- Generative model
  - Predict acoustic frames from text
SPSS

- **ASR vs SPSS**
  - Similar techniques but not the same

- **Model training techniques**
  - Alignment, and cluster features
  - MLLR (adaptation from multi-speaker models)

- **Model improvement techniques**
  - Minimum generation error
  - Label optimization

- **Parameterization techniques**
  - MFCC, LSP, STAIGHT, HSM
  - Excitation modeling techniques
SPSS Goals

- **Require optimal parameterization that**
  - Is derivable from speech
  - Can generate high quality speech
  - Is predictable from text

- **Candidates**
  - Spectral, F0, excitation
  - Formants, nasality, aspiration
  - Articulatory features
HTS (NITECH)
- Based on HTK
- Predicts HMM-states
- (Default) uses MCEP and MLSA filter
- Supported in Festival

Clustergeren (CMU)
- No use of HTK
- Predicts Frames
- (Default) uses MCEP and MLSA filter
- More tightly coupled with Festival
Building Synthetic Voices

The “standard” voice requires …
- A phone set
- Pronunciations:
  - Lexicon/letter-to-sound rules
- Phonetically and prosodically balanced corpus
  - Spoken by a good speaker
- Text analysis:
  - Number, symbol expansion, etc
- Prosodic modeling
  - Phrasing, intonation, duration etc
- Waveform generation
  - Diphones, unit selection, parametric synthesis
- Something else that is hard:
  - No vowels (Arabic), no word segmentation, number declensions
Designing a good corpus

- From a large set of text
  - Select “nice” utterances
  - 5 to 15 words, easy to say
  - All words in lexicon, no homographs
- Convert text to phoneme strings
  - Possibly with lexical stress, onset/coda, tone etc
- Select utterances that maximize di/triphone coverage
- Looking for around 1000 utterances
- Can seed initial data with “domain” data
- CMU ARCTIC databases
  - 7 x single speaker English DBS
  - 1200 phonetically balanced utterances
Hard Synthesis Problems

- Text Normalization
- Intonation modeling
  - Intonation evaluation
- Style modeling
  - Choosing the right style
  - Evaluating the result
Text Normalization

- **Finding the words**
  - Tokenizing, homograph disambiguation etc
  - "$1.25" vs "$1.25 million" vs "$1.25 song"

- **Very large number of rare events**

- **Formalized systems exist**
  - Trained from data, optimized and out-of-date

- **Long term updated hacks rule systems**

- **ML Challenge**
  - Such a problem cannot be done by machine learning
Intonation Modeling

- **Accents, Phrases and F0**
  - Lots of statistical models available
  - Lots of “objective” measures:
    - RMSE, Correlation
  - No good subjective measures

- **Listening tests**
  - Natural Intonation: good
  - Naïve intonation: bad
  - Various cute models for intonation: meh
Improving Understanding

- **Take reading comprehension stories**
  - For children’s reading tests, or TOEFL

- **Synthesis with:**
  - Natural Intonation
  - Naïve models
  - Various cute models

- **Human listening tests**
  - Answer questions about stories
  - Best system: Naïve models 😞
Style Modeling

- **Classic Emotion Modeling**
  - Happy, sad, angry and neutral
  - But no one needs that

- **Style Modeling**
  - Polite, command, empathic

- **Style usage**
  - When can it be used?
  - How much should be used?
Dialog with Style

- **Record human-human dialog**
  - Label dialog states:
    - Implicit confirmation, corrections, discourse markers

- **Build dialog state sensitive voice**
  - Using dialog state in features

- **Must be closely integrated into SDS**
  - Timing, dialog state appropriate

- **But how do you test it?**
Voice Transformation

- Collect small amount of data
  - 50 utterances
- Adapt existing voice to target voice
- Adaptation: What makes a voice:
  - Lexical choice
  - Phonetic variation
  - Prosody
  - Spectral/vocal tract/articulatory movement
  - Excitation mode
- Use articulatory modeling for transformation (Toth)
Voice Transformation

- Festvox GMM transformation suite (Toda)

<table>
<thead>
<tr>
<th>awb</th>
<th>bdl</th>
<th>jmk</th>
<th>slt</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
</tr>
<tr>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
</tr>
<tr>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
</tr>
<tr>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
<td>🎧</td>
</tr>
</tbody>
</table>
Applications

- Speech output is only one component
- Need to integrate with larger applications
  - Spoken Dialog Systems
  - Speech-to-Speech Translation Systems
  - Talking Heads
  - Conversational participants
  - Information delivery
Conclusions

• Synthesis has improved
  – But there is still much to do
  – Isolated sentences are clear …
  – … But conversational speech still in the future

• Speech Systems must adapt
  – To their usage
  – And their funding conditions

• But we can always fall back on our talents