Text-to-Speech Synthesis

Bernd Möbius
Language Science and Technology
Saarland University

Lecture 5
June 18, 2020

Unit Selection Synthesis

(Materials on pp. 2, 4-8, 10, 11 based on Simon King's keynote speech at ICPhS 2015)
Corpus-based synthesis

- 2 modern corpus-based synthesis methods
  - **concatenation**: constructing a synthetic utterance from units of recorded speech (unit selection synthesis)
  - **generation**: using a sequence of models to generate a synthetic utterance (statistical-parametric synthesis)
**Unit selection synthesis: general procedure**

- Dynamic selection of units at synthesis run-time
  - "The best solution to the synthesizer problem is to avoid it." [Carlson & Granström, 1991]
  - overcome restrictions by a fixed unit inventory
  - unit inventory: large corpus of recorded natural speech
  - select the smallest number of the longest units covering the target phone sequence
  - variable unit size (segments, syllables, words, ...)
  - reduce perceptual impression of lack of naturalness caused by number of concatenations and signal processing
Unit selection synthesis

- Creating a new word

train
Unit selection synthesis

- Creating a new word

train
Unit selection synthesis

- Creating a new word

train

peas
Unit selection synthesis

- Creating a new word

train

peas
Unit selection synthesis

- Creating a new word

train

peas

trees
Paradigm change

- Inventory construction off-line $\Rightarrow$ run-time unit selection
  - preserve natural speech as much as possible
  - *ideal world*: target utterance available in corpus
  - *unfortunately*: ideal case is extremely improbable, due to complexity/combinatorics of language and speech
  - however, longer units may be available in corpus
  - most extreme strategy (CHATR, Black & Taylor 1994 ...)
    - no modification by signal processing
    - listener will tolerate occasional glitches, if overall synthesis quality approaches natural speech
Unit selection synthesis

- creating a new word, like building a car from second-hand parts
- try to select parts that fit together nicely
Unit selection synthesis

- sometimes the parts don't fit as nicely as desired
- most of the times the concatenation is not smooth
- sometimes you can hear the joins (glitches)
- other, more subtle, artifacts will occur too
Unit selection based on cost functions

- Minimize two cost functions, simultaneously and globally (viz. for the entire utterance)
  - **target cost** (unit distortion): how suitable is the candidate?
    
    *Is this a nice driver-side front door?*
  
  - **concatenation cost** (join cost, continuity distortion): how smooth is the concatenation with adjacent units?
    
    *How smoothly does this door connect to other parts?*
Cost functions

- **target cost \((C^t)\)**
  - distance of candidate unit from target specification
  - computed at run-time
  - based on features that are inferred from input text

- **concatenation cost \((C^c)\)**
  - distance between two candidates at concatenation
  - computable off-line, but usually at run-time
  - based on all features in corpus annotation
Selection algorithm

- Minimize $C^t$ and $C^c$ [Hunt & Black 1996]

sequence of target units

lattice of candidate units
Selection algorithm

- Minimize $C^t$ and $C^c$ [Hunt & Black 1996]

Diagram:

- Nodes labeled $m$, $a$, $n$, $m_{123}$, $a_{55}$, $n_{219}$
- Arrows indicate target costs and concatenation costs
Lattice of candidate units

- Every unit in corpus is represented by a state in a state-transition network
  - state cost = target cost (unit distortion)
  - transition cost = concatenation cost (continuity dist.)
- Similar to HMM-based speech recognition, but
  - ASR uses probabilistic models
  - unit selection uses cost functions
- Unit selection algorithm selects an optimal sequence of units by finding a path through the state-transition network that jointly minimizes target and join costs [Hunt & Black 1996]
Example: Word-based unit selection

- Target utterance: *I have time on Monday.*
- Step 1: tabulate all candidate words for target utterance

<table>
<thead>
<tr>
<th>I</th>
<th>have</th>
<th>time</th>
<th>on</th>
<th>Monday</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>have</td>
<td>time</td>
<td>on</td>
<td>Monday</td>
</tr>
<tr>
<td>I</td>
<td>have</td>
<td>time</td>
<td>on</td>
<td>Monday</td>
</tr>
<tr>
<td>I</td>
<td>have</td>
<td>on</td>
<td></td>
<td>Monday</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>on</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example: Word-based unit selection

- Target utterance: *I have time on Monday.*
- Step 2: create directed graph with start and end nodes
Example: Word-based unit selection

- Target utterance: *I have time on Monday.*
- Step 3: selection of units by optimal (cheapest) path
Cost functions: Features

- Mismatch between selection criteria in target vs. corpus will produce costs and result in suboptimal selection.
- Commonly used features in target cost:
  - position in phrase (initial, medial, final)
  - duration (actual vs. predicted by duration model)
  - sentence mode (declarative, interrogative, continuing)
  - degree of reduction
- Commonly used features in concatenation cost:
  - adjacency in corpus (zero cost)
  - left and right phonetic context
  - $F_0$ (and possibly intensity) continuity
Cost functions

- Costs are computed as weighted sums of costs incurred on individual features
  \[ C = w_1c_1 + w_2c_2 + w_3c_3 + \ldots \]

- Feature weights are set by expert or trained on data
  - finding optimal relative weights of features is still an open problem

- Optimal match on feature will incur zero cost
  - in join cost, this will prefer sequences of candidates that are adjacent in corpus, and thus longer units
Features: annotation and use in selection

- Off-line:
  - annotation of speech corpus (feature vectors)
    - phone identities and phone boundaries
    - prosodic features (tones, accents, stress)
    - positional features (in phrase, word, syllable)
  - possibly: concatenation costs

- On-line = at synthesis run-time
  - target specifications (target feature vectors)
  - target costs
  - typically: concatenation costs
Speech corpus design and size

- Quality of speech corpus (recordings, annotation, coverage) has tremendous effect on synthesis quality
- Corpus size is single most important quality factor
- Some data points:
  - IBM/Cambridge: ~60 min. (ASR corpora)
  - CHATR: phonetically balanced sentences, radio news, isolated sentences: 40 min. (Eng.), 20 min. (Jap.)
  - "bring a novel ... of their own choice" [Campbell 1999]
  - AT&T 1999: news stories and system prompts, ~2 hrs.
  - SmartKom: open+closed domains, 160 min.
  - typical corpus size today: 10++ hrs.
Speech corpus design and size

- Quality of speech corpus (recordings, annotation, coverage) has tremendous effect on synthesis quality
- Careful design of text materials essential
  - overall objective: consistently high quality
  - account for all relevant phone variations
  - complete coverage by diphones
  - various text genres (news, dialog, prompts)
  - various speaking styles (prosody, affective)
  - automatic annotation with manual verification
    - small corpora with perfect annotation yield better quality than large corpora without verification
Unit selection synthesis: Summary

- Synthesis by re-sequencing and concatenating units selected at run-time from corpus of natural speech
  + facilitates long units without concatenation
  + reduces need for signal processing
  + preserves natural speech waveforms
  - tends to produce unstable, unpredictable quality
  - inflexible w.r.t. speaking style and speaker voice

- Standard synthesis technique in the 2000s
  - in competition with HMM-based synthesis (statistical parametric speech synthesis)
Essential content: unit selection synthesis

What are the differences between unit selection and previous concatenative synthesis methods (e.g., diphone synthesis)?

How does unit selection work and what types of acoustic units are available?

Discuss problems and solutions pertaining to the design of acoustic unit inventories for unit selection synthesis (e.g., corpus design, domain coverage).