Speech Synthesis
Text-to-speech (TTS)

• Definition: a text-to-speech system must be
  • Able to read any text
  • Intelligible
  • Natural sounding

• The first of these puts a constraint on the method we can choose:
  • playback of whole words or phrases in not a solution

• The second is actually closer to being a ‘solved problem’ than the third

• A generation task
  • although not completely clear what objective function we are optimising
From text to linguistic specification

phrase initial

pitch accent

phrase final

sil dh ax k ae t s ae t sil

"the cat sat"

DET NN VB

((the cat) sat)
From linguistic specification to a waveform

• **Concatenation** builds up the utterance from units of recorded speech:

• **Generation** uses a model to generate the speech:

  could be a sequence of HMMs, or a single DNN
Synthetic speech created from audiobooks

Audio credits: Speech and Hearing Research Center, Peking University
DNN speech synthesis
Training

Error: \( (o'_t - o_t)^2 \)

Back-propagation
<table>
<thead>
<tr>
<th></th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Natural</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HMM-GV</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DNN-LSTM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BN-DNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TE-DNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TE-BN-DNN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Speech Synthesis: open problem 1

From **input feature engineering** (traditional NLP and knowledge sources) to

learned-from-data linguistic features
Standard text processing pipeline

Front end

- tokenize
- POS tag
- LTS
- Phrase breaks
- intonation

individually learned from labelled data
Text processing pipeline

- A chain of **processes**
- Each process is performed by a **model**
- These models are independently trained in a **supervised** fashion on annotated data
Example process 1

- Part-of-speech tagger
- Accuracy is very high
- **But**
  - trained on **annotated** text data
  - **categories** are designed for text, not speech
Example process 2

- Pronunciation model
  - dictionary look-up, plus
  - letter-to-sound model
- But
  - need deep knowledge of the language to design the phoneme set
- human expert must write dictionary

This sequence is the annotated training data for our letter-to-sound predictor.
Example process 3

- Phrase-break prediction is the annotated training data for our phrase break predictor.
- This sequence is the annotated training data for our phrase break predictor.
- But trained on annotated spoken data.
- Therefore very small training set.
Representing linguistic features

- **Encoding**
  - 1-of-N for phoneme identity, POS, etc
  - binary partitions of the space, e.g. “is this a vowel”
  - positional features
    - within syllable, word, phrase

- **Representing context**
  - include previous & next phonemes, etc
  - some features span the current utterance

- **Problems**
  - sparsity (mostly zeros)
  - noise (errors in linguistic processing)
  - relevance (not all features are predictive of speech)
Learning embeddings of features
Stacking up more context
Speech Synthesis: open problem 2

From frame-by-frame prediction to trajectory generation
Frame-by-frame prediction

Vocoder parameters

$\mathbf{x}_{t-1}$

Linguistic features

Vocoder parameters

$\mathbf{x}_t$

Linguistic features

Vocoder parameters

$\mathbf{x}_{t+1}$

Linguistic features
Inconsistency

Smoothed parameter trajectories

Trajectory generation

Training

Generation

\( C' \)
Trajectory generation

Training

Generation

$C$

$C'$

Error: $(C' - C)^2$

Back-propagation
Speech Synthesis: open problem 3

From **speaker-dependent** speech synthesis to adaptable and controllable models

Lots of work already on this in the HMM framework, but still remains an open problem for DNNs
Different ways to adapt the DNN

Gender code

i-vector

LHUC

Feature mapping

Vocoder parameters

Vocoder parameters

Linguistic features

Gender code
Speech Synthesis: open problem 4

From \textbf{output feature engineering} (speech signal modelling, a.k.a vocoding) to learned-from-data speech generation
What to predict?

Vocoder parameters

$h_4$

$h_3$

$h_2$

$h_1$

$x_t$

Linguistic features
Direct waveform generation?