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 is 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

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

1 paragraph example

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

Vocoder parameters

\[ o_1, o_2, \ldots, o_t, \ldots, o_T \]

Mapping

Linguistic features

\[ x_1, x_2, \ldots, x_t, \ldots, x_T \]

Vocoder parameters

\[ o_t \]

\[ h_1 \]

\[ h_2 \]

\[ h_3 \]

\[ h_4 \]

Linguistic features

\[ x_t \]
Training

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

Back-propagation
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

*linguistic specification*

*text*

*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

Example text:

- ADVOCATING  AE1 D V AH0 K EY2 T IH0 NG
- ADVOCATION  AE2 D V AH0 K EY1 SH AH0 N
- ADWEEK  AE1 D W IY0 K
- ADWELL  AH0 D W EH1 L
- ADY  EY1 D IY0
- ADZ  AE1 D Z
- AE  EY1
- AEGEAN  IH0 JH IY1 AH0 N
- AEGIS  IY1 JH AH0 S
- AEGON  EY1 G AA0 N
- AELTUS  AE1 L T AH0 S
- AENEAS  AE1 N IY0 AH0 S
- AENEID  AH0 N IY1 IH0 D
- AEQUITRON  EY1 K W IH0 T R AA0 N
- AER  EH1 R
- AERIAL  EH1 R IY0 AH0 L
- AERIALS  EH1 R IY0 AH0 L Z
- AERIE  EH1 R IY0
- AERIEN  EH1 R IY0 AH0 N
- AERIENS  EH1 R IY0 AH0 N Z
- AERITALIA  EH2 R IH0 T AE1 L Y AHC
- AERO  EH1 R OW0
Example process 2

AERIALS

EH1 R IY0 AH0 L Z

Example process 2 text processing pipeline:
- Front end
  - Tokenize
  - POS tag
  - LTS
  - Phrase breaks
  - Intonation

Individually learned from labelled data.
This sequence is the annotated training data for our letter-to-sound predictor.

<table>
<thead>
<tr>
<th>A</th>
<th>–</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>EH1</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>I</td>
<td>IY0</td>
</tr>
<tr>
<td>A</td>
<td>AH0</td>
</tr>
<tr>
<td>L</td>
<td>L</td>
</tr>
<tr>
<td>S</td>
<td>Z</td>
</tr>
</tbody>
</table>
Example process 3

- Phrase-break prediction
  - binary classifier using POS sequence as input
- But
  - trained on annotated spoken data
  - therefore very small training set
Example process 3

Text processing pipeline

- Break!

Front end
- tokenize
- POS tag
- LTS
- Phrase breaks
- Intonation

individually learned from labelled data

Break!
Example process 3

Text processing pipeline

9

Front end

LTS

Linguistic specification

Phrase breaks

POS tag

Tokenization

Individually learned from labelled data
Example process 3

This sequence is the annotated training data for our phrase break predictor.
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

\( o_{t-1} \)

\[ x_{t-1} \]

Linguistic features

Vocoder parameters

\( o_t \)

\[ x_t \]

Linguistic features

Vocoder parameters

\( o_{t+1} \)

\[ x_{t+1} \]

Linguistic features
Inconsistency

Smoothed parameter trajectories

Trajectory generation

Training

Generation
Trajectory generation

Training

$C$

$C'$

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

Generation

Trajectory generation

$h_1$, $h_2$, $h_3$, $h_4$

$o_1$, $o_2$, $o_t$, $o_T$

$x_1$, $x_2$, $x_t$, $x_T$

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
Speech Synthesis: open problem 4

From **output feature engineering** (speech signal modelling, a.k.a vocoding) to

**learned-from-data speech generation**
What to predict?

Vocoder parameters

\[ o_t \]

\[ h_4 \]
\[ h_3 \]
\[ h_2 \]
\[ h_1 \]

Linguistic features

\[ x_t \]
Direct waveform generation?