CS 224S / Linguist 285
Spoken Language Processing

Andrew Maas | Stanford University | Spring 2025

Lecture 4: TTS Waveform Synthesis

Outline

- Recipe for building great TTS
- Unit selection synthesis
- Creating audio waveforms
- Setting up parametric synthesis. What to predict?

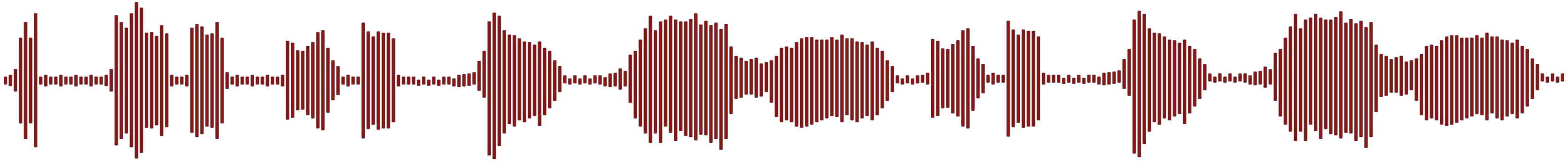
Recipe for Building Great TTS

The Two Stages of TTS

```
PG&E will file schedules on April 20th
```

1. Text Analysis: Text into intermediate representation:

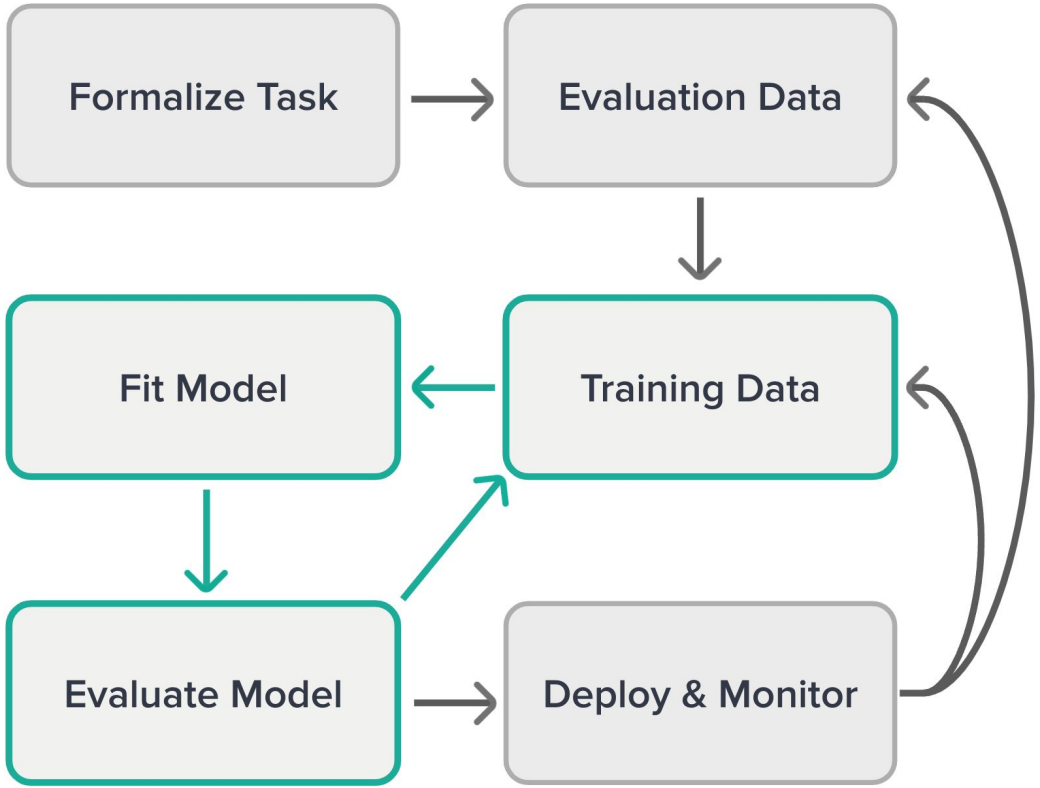
2. Waveform Synthesis: From the intermediate representation into waveform



Modern TTS relies on machine learning

- Match training data + system architecture with planned usage
- High quality training datasets with broad coverage to achieve desired voice
- Leverage existing models as starting point if applicable
- *So, how do we build a great ML system that uses audio from one or more people?*

Iteratively developing ML-based TTS systems



What is the Recipe For Building Great TTS?

- **Evaluation & measurement**
  - Choose criteria (natural, emotive?)
  - Set up **human** evaluation listening tests
- **Data collection**
  - TTS acoustic quality limited by collected data
  - Require emotional range, expressiveness
- **Modeling**
  - Deep learning systems work best
  - Concatenative systems easier to build fast
  - Design controllable interface for developer

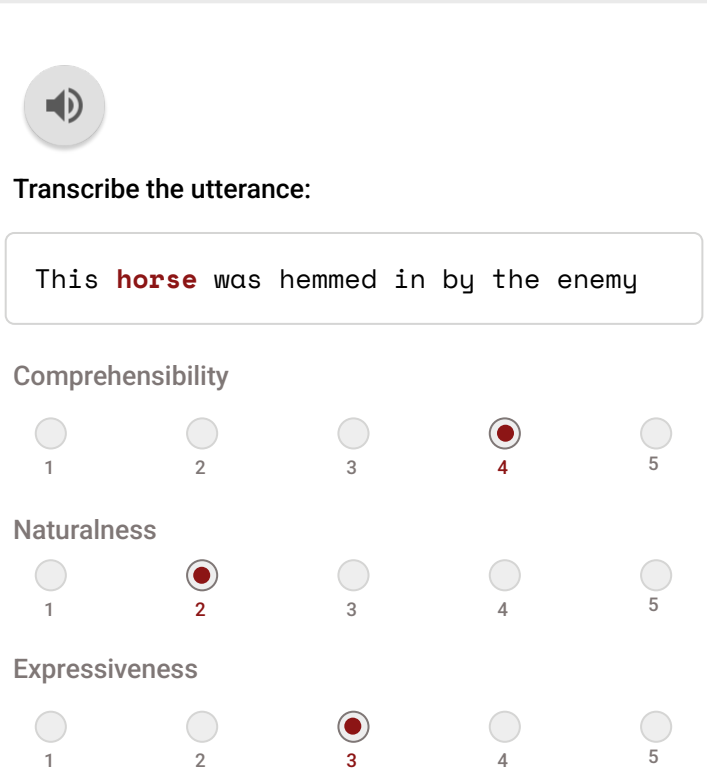
Evaluation of TTS

- **Evaluation of TTS generally requires humans!**
  - Listening test paradigm. Listen to example utterances, rate various aspects (naturalness, intelligibility, friendliness, expressiveness, etc.). Scale of 1-5
  - Mean opinion score (MOS). Average of ratings
  - AB Tests (prefer A, prefer B) (preference tests)
- **Intelligibility Tests**
  - Did the human hear the correct thing? Can test task completion, writing what was said, or simply rating
- **Overall Quality Tests**
  - A/B preference test vs human narrator is “ceiling”

Mean Opinion Score

Using crowdsourced ranking of synthesis results based on:

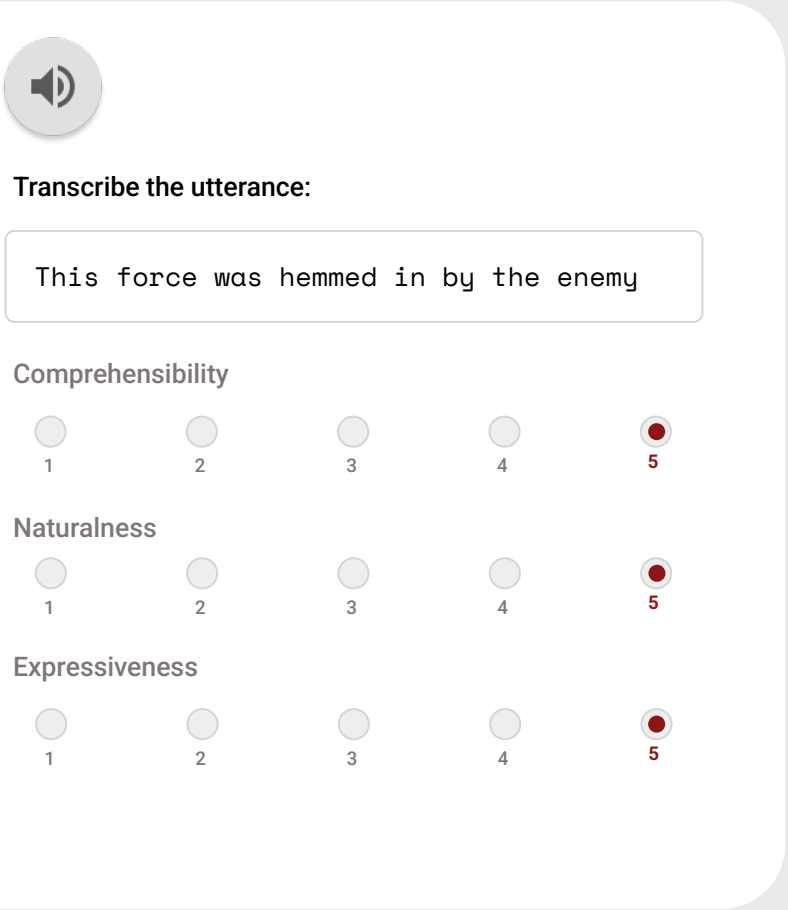
- **Intelligibility**
  - Usually quantified objectively via transcription
- **Comprehensibility**
  - How easy is it to understand a particular utterance
- **Naturalness**
  - How natural does the utterance sound
- **Expressiveness**
  - How well does the intonation match the substance of the utterance



Mean Opinion Score

Using crowdsourced ranking of synthesis results based on:

- **Intelligibility**
  - Usually quantified objectively via transcription
- **Comprehensibility**
  - How easy is it to understand a particular utterance
- **Naturalness**
  - How natural does the utterance sound
- **Expressiveness**
  - How well does the intonation match the substance of the utterance

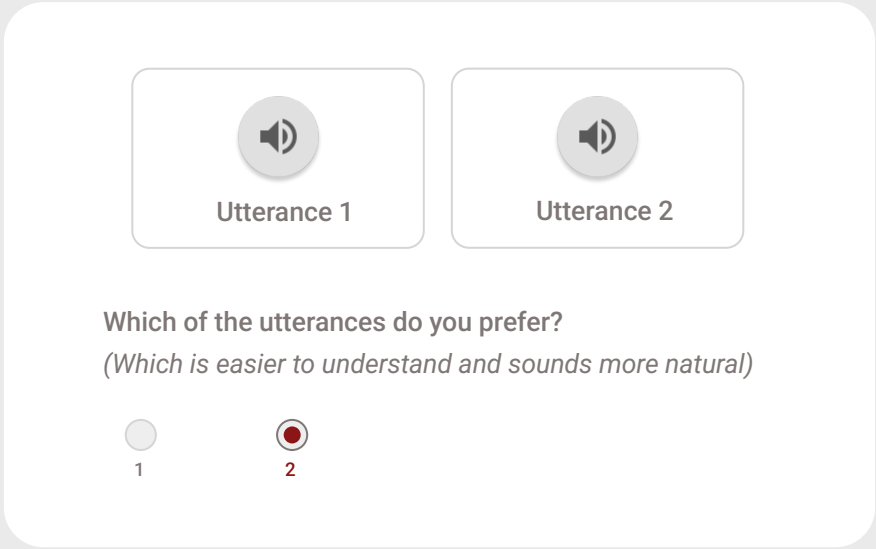


A/B Testing

Using crowdsourced selections to elicit direct preferences for TTS settings

Often much more reliable than comparing ratings of systems evaluated in isolation

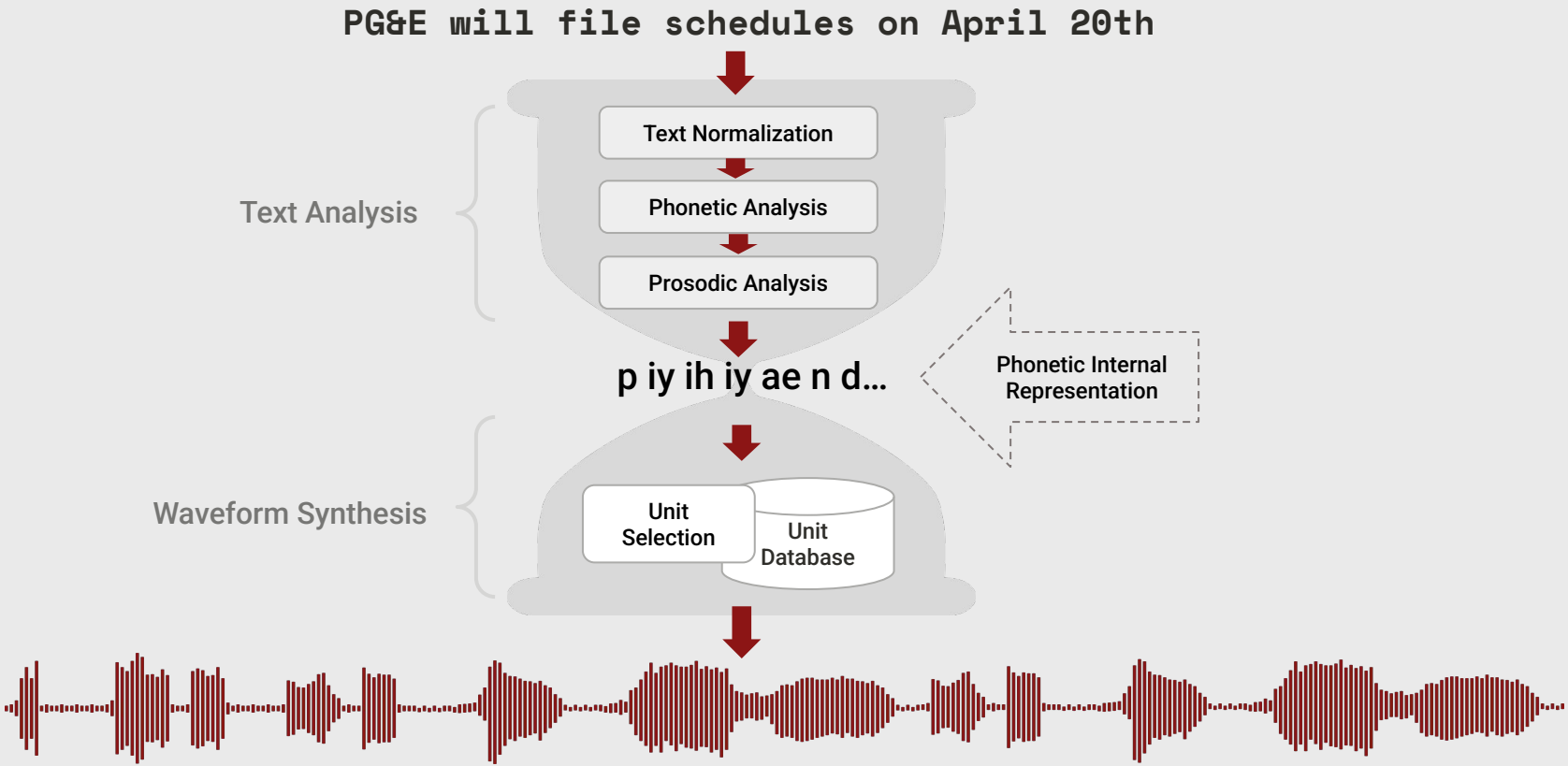
Answers direct questions about differences in model versions / experiment hypothesis



Evaluation of TTS

- **Diagnostic Rhyme Test (DRT)**
  - Humans do listening identification choice between two words differing by a single phonetic feature
    - Voicing, nasality, sustentation, sibilation
  - 96 rhyming pairs
  - Veal/feel, meat/beat, vee/bee, zee/thee, etc
    - Subject hears “veal”, chooses either “veal or “feel”
    - Subject also hears “feel”, chooses either “veal” or “feel”
  - % of right answers is intelligibility score.

Data Collection for TTS



Data Collection for TTS

- **Great acoustic quality**
  - At least 16 kHZ
  - Good microphone
  - Minimal background noise (including page turning and breathing!)
- **Emotional and phonetic range to match application**
  - System will clone accent of single speaker
  - Must collect emotional speech if TTS needs to produce it
  - Read vs conversational speech is different. Simulate human-human conversations with role play possibly
- **Enough data**
  - ~10 hours might be enough for read speech (single speaker)
  - Transfer learning enables some data sharing

Recording for Google Assistant (unit selection)

Great recording conditions, attention to prosody, units for common phrases

More important for unit selection systems, but data quality can be limiting factor for modern TTS

Link to video

Video link

Tacotron2 Architecture

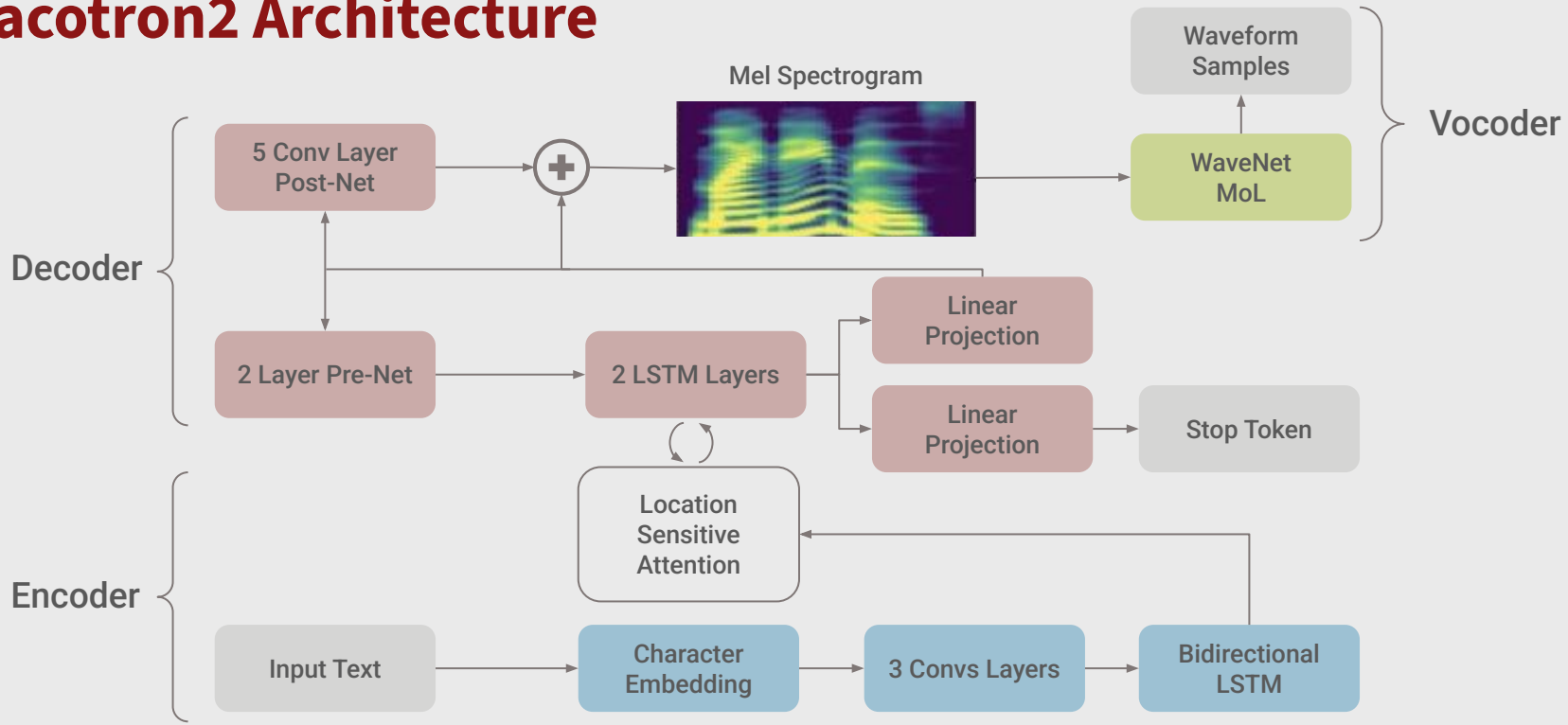


Figure: Tacotron2 architecture: An encoder-decoder maps from graphemes to mel spectrograms, followed by a vocoder that maps to wavefiles.Shen et al. (2018)

Concatenative Waveform Synthesis

A quick, incomplete introduction

- **Create a single speaker database of speech ‘units’ (typically diphones up to whole words)**
- **Unit selection search**
- **Joining units**

Unit Selection Intuition

What does “best” unit mean?

- **Target cost:** closest match to the target description, in terms of
  - Phonetic context
  - F0, stress, phrase position
- **Join cost:** best join with neighboring units
  - Matching formants + other spectral characteristics
  - Matching energy
  - Matching F0

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^{target}(t_i, u_i) + \sum_{i=2}^{n} C^{join}(u_{i-1}, u_i)$$

Join (Concatenation) Cost

- Measure of smoothness of join
- Measured between two database units (target is irrelevant)
- Features, costs, and weights
- Comprised of k subcosts:
  - Spectral features
  - F0
  - Energy
- Join cost:

$$C^j(u_{i-1},u_i) = \sum_{k=1}^{p} w_k^j C_k^j(u_{i-1},u_i)$$

Slide: Paul Taylor

Total Costs

- Hunt and Black 1996
- We now have weights (per phone type) for features set between target and database units
- Find best path of units through database that minimize:

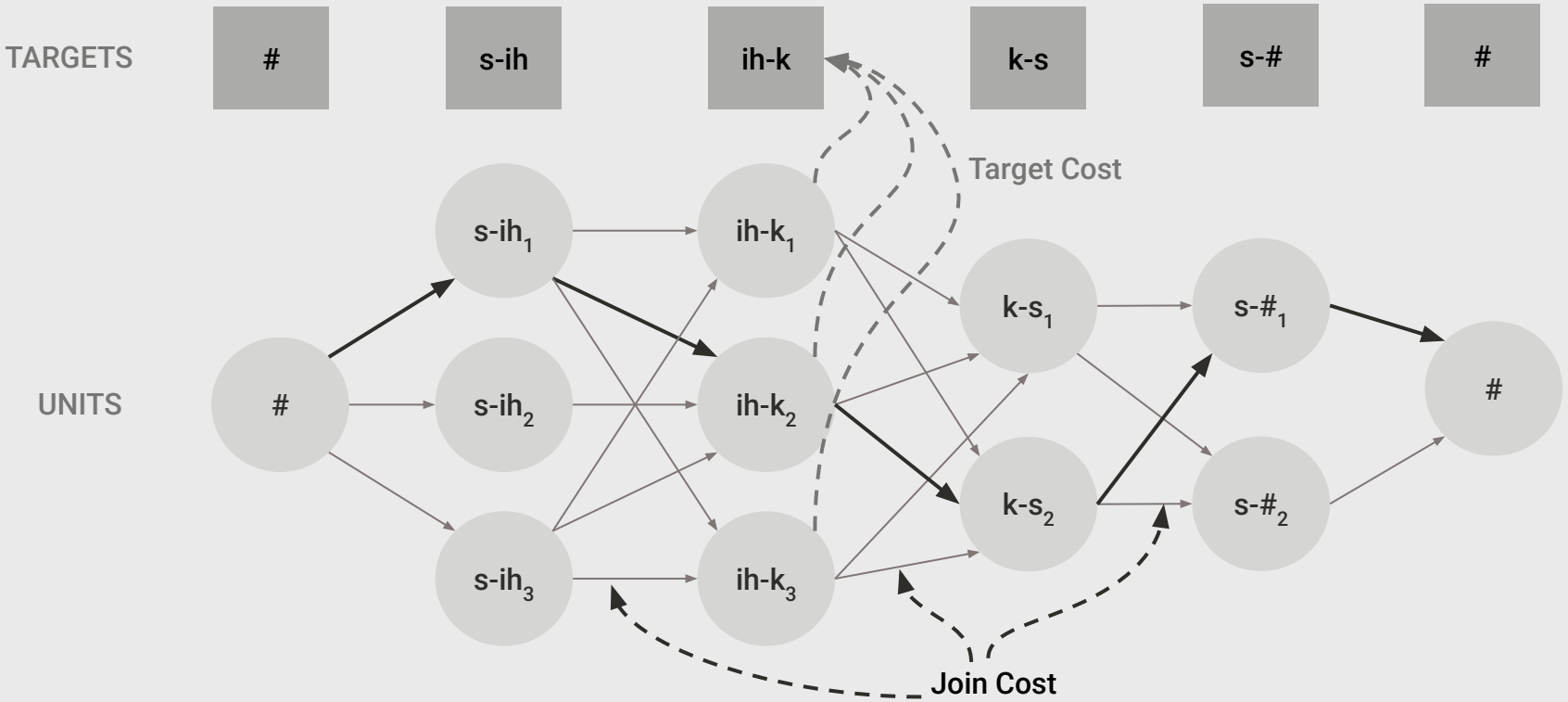
$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^{target}(t_i, u_i) + \sum_{i=2}^{n} C^{join}(u_{i-1}, u_i)$$

$$\hat{u}_1^n = \underset{u_1, \ldots, u_n}{\operatorname{argmin}} C(t_1^n, u_1^n)$$

- Standard problem solvable with Viterbi search with beam width constraint for pruning

Slide: Paul Taylor

Unit Selection Search



Waveform Synthesis

Given:

- String of phones
- Prosody: *We need to generate unique prosody for new utterances and impose it*
  - Desired F0 for entire utterance
  - Duration for each phone
  - Stress value for each phone, possibly accent value

Generate:

- Waveforms

F0 Generation

- By rule
- By linear regression / machine learning
- Some constraints
  - By accents and boundaries
  - F0 declines gradually over an utterance (“declination”)

Speech as Short Term Signals: Carefully joining units

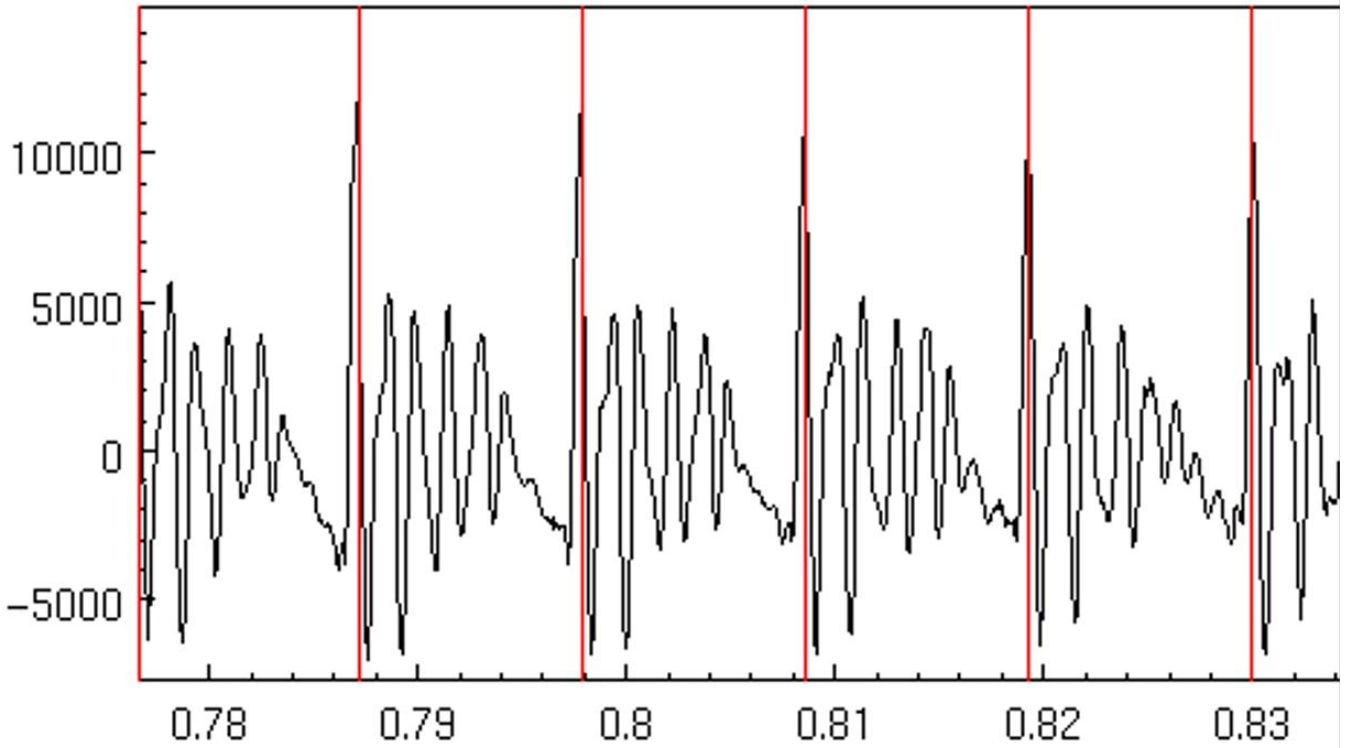
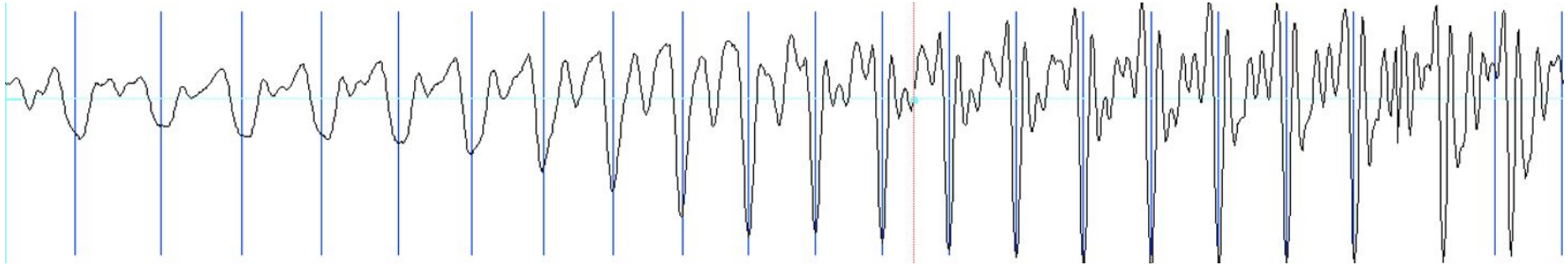


Figure: Alan Black

Epoch-labeling

- An example of epoch-labeling using “SHOW PULSES” in Praat:



Duration Modification

- Duplicate/remove short term signals (align to epoch boundaries for cuts)

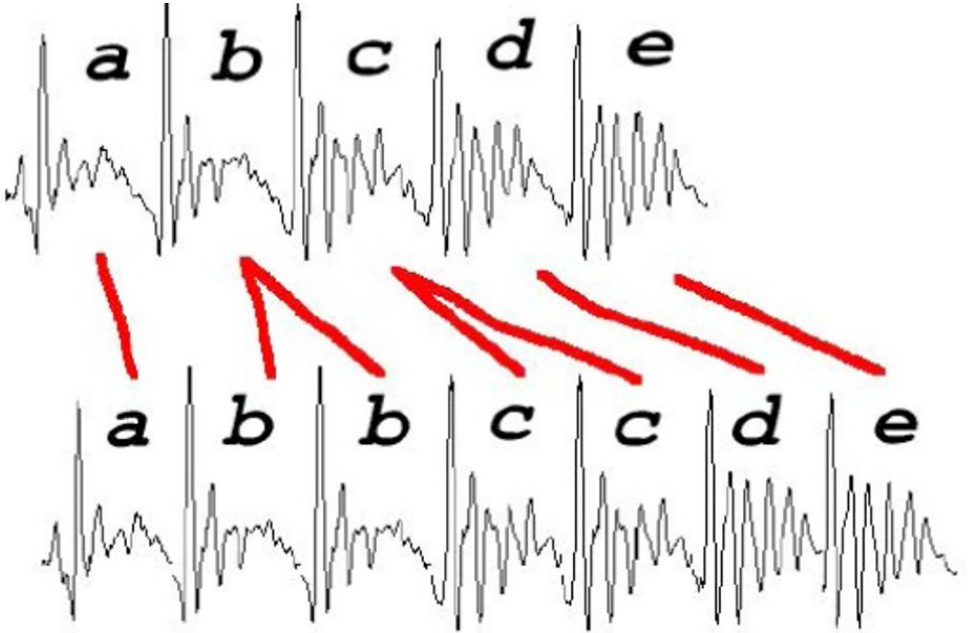


Figure: Richard Sproat

Pitch Modification

- Move short-term signals closer together/further apart

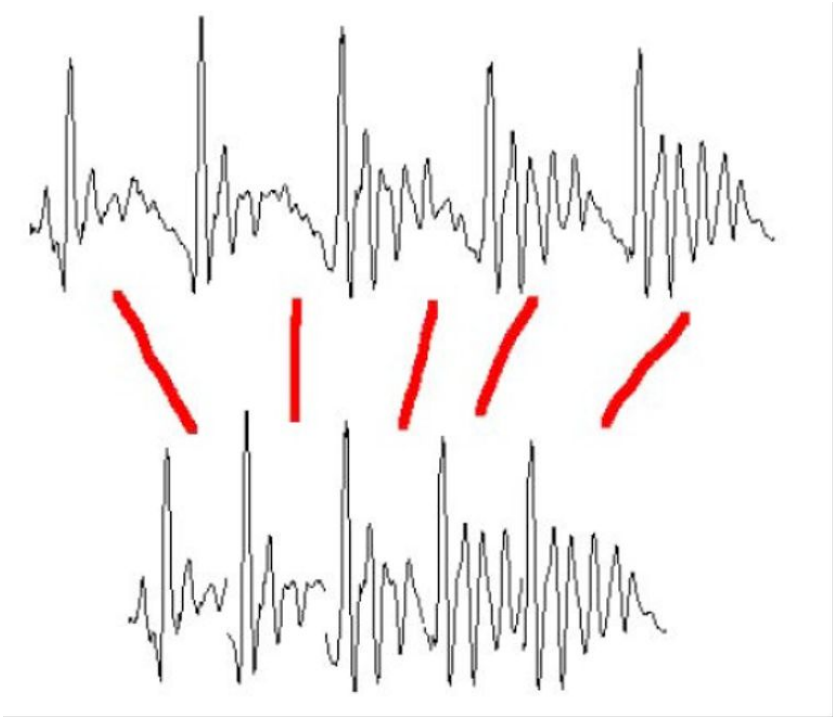


Figure: Richard Sproat

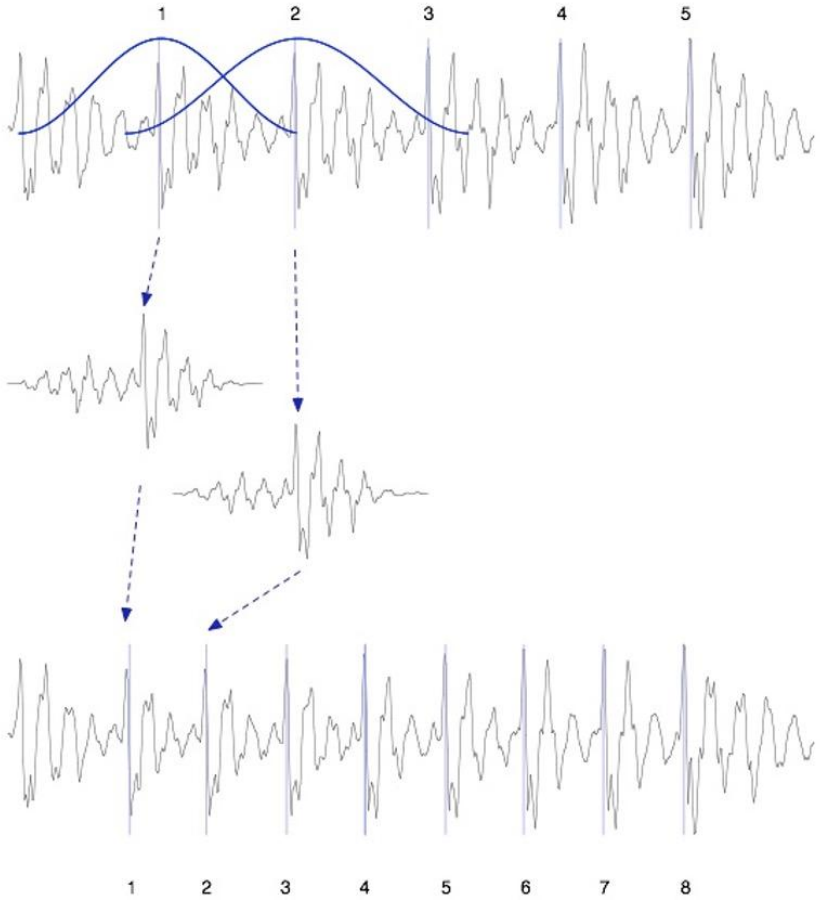
TD-PSOLA ™

Time-**D**omain (Windowed)

Pitch-**S**ynchronous

Over**l**ap-and-**A**dd

- Efficient
- Wide range of Hz
- Join units of any size



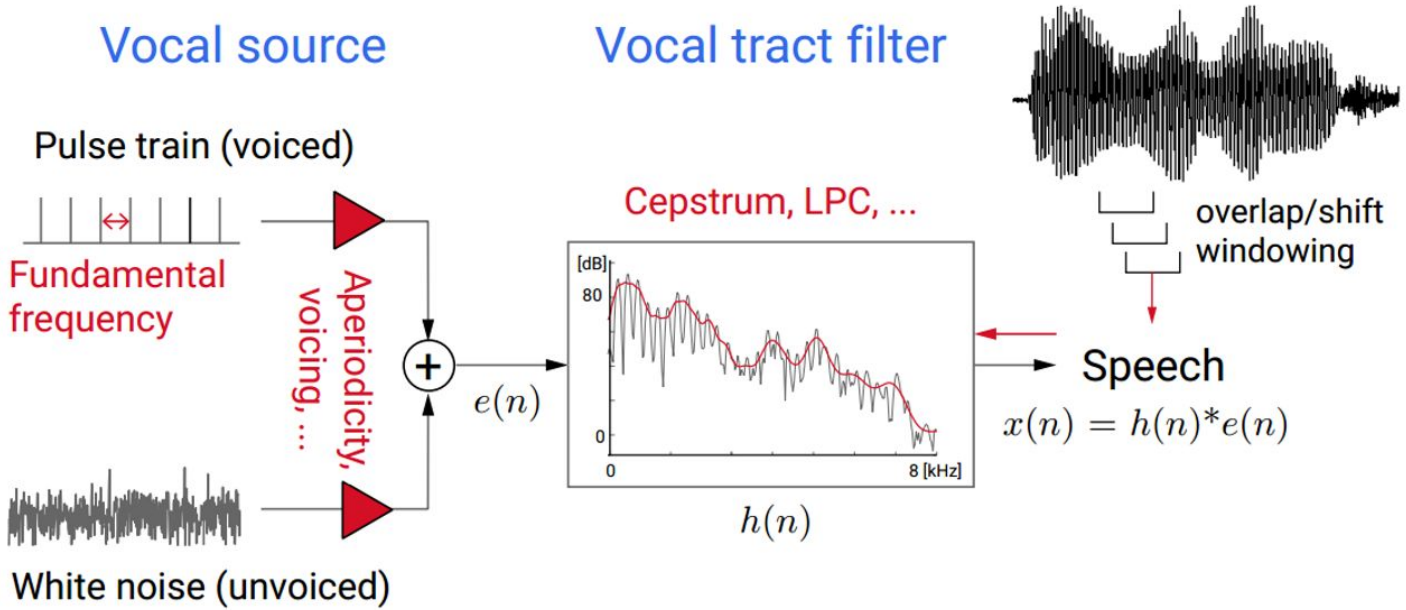
What to Predict in Parametric Synthesis

Key Questions in Parametric Synthesis

- What parameters do we predict?
  Usually MFCCs for spectrum, log F0, voicing/excitation
- How do we combine them (vocoding)?
  Exact parameterization and combining them well reduces robotic buzzy effects
- How do we make predictions? Choice of HMM, machine learning approaches.
  Less important than the vocoding/combination issues
- For a chosen input/output representation for TTS, how can you obtain high quality labels for your representation?

Synthesis with Source-filter Model

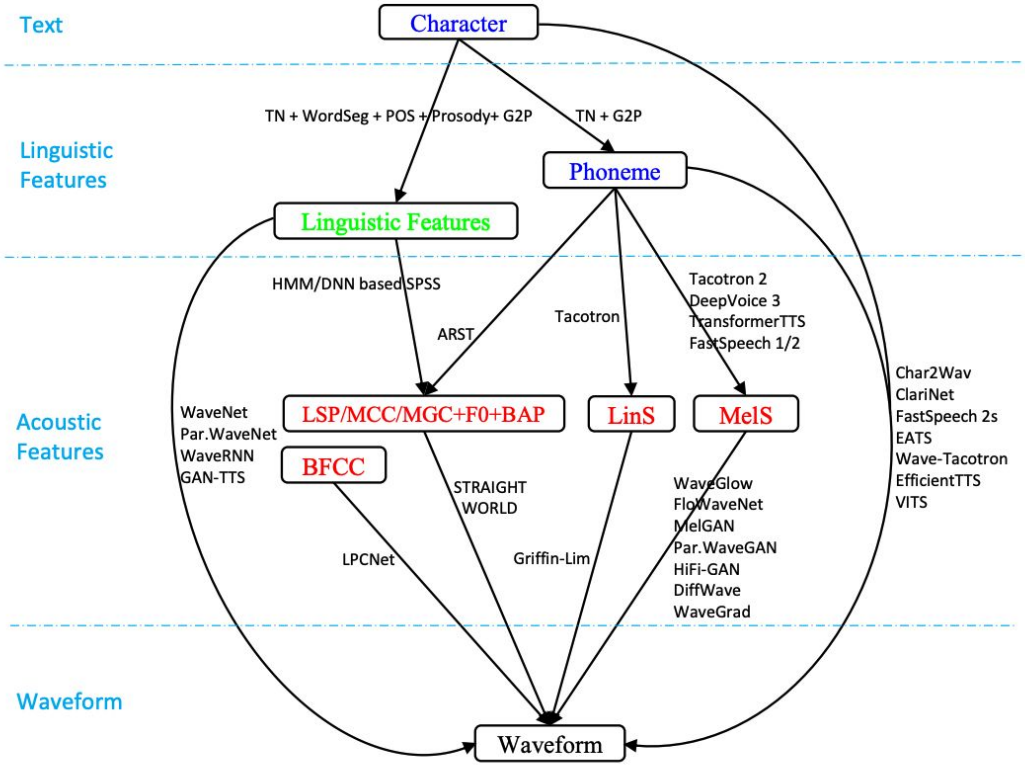
Piece-wise stationary, source-filter generative model $p(x \mid o)$



The design space of intermediate representations

Different models choose different internal representations from text to audio.

Choice of representation impacts modeling, data collection/annotation, and controllability of resulting system



(b) The data flows from text to waveform.

([Tan, Qin, Soong & Liu, 2021](#))

What Does the HMM* Produce?

***We don't use HMM's anymore, but any ML system has the same question**

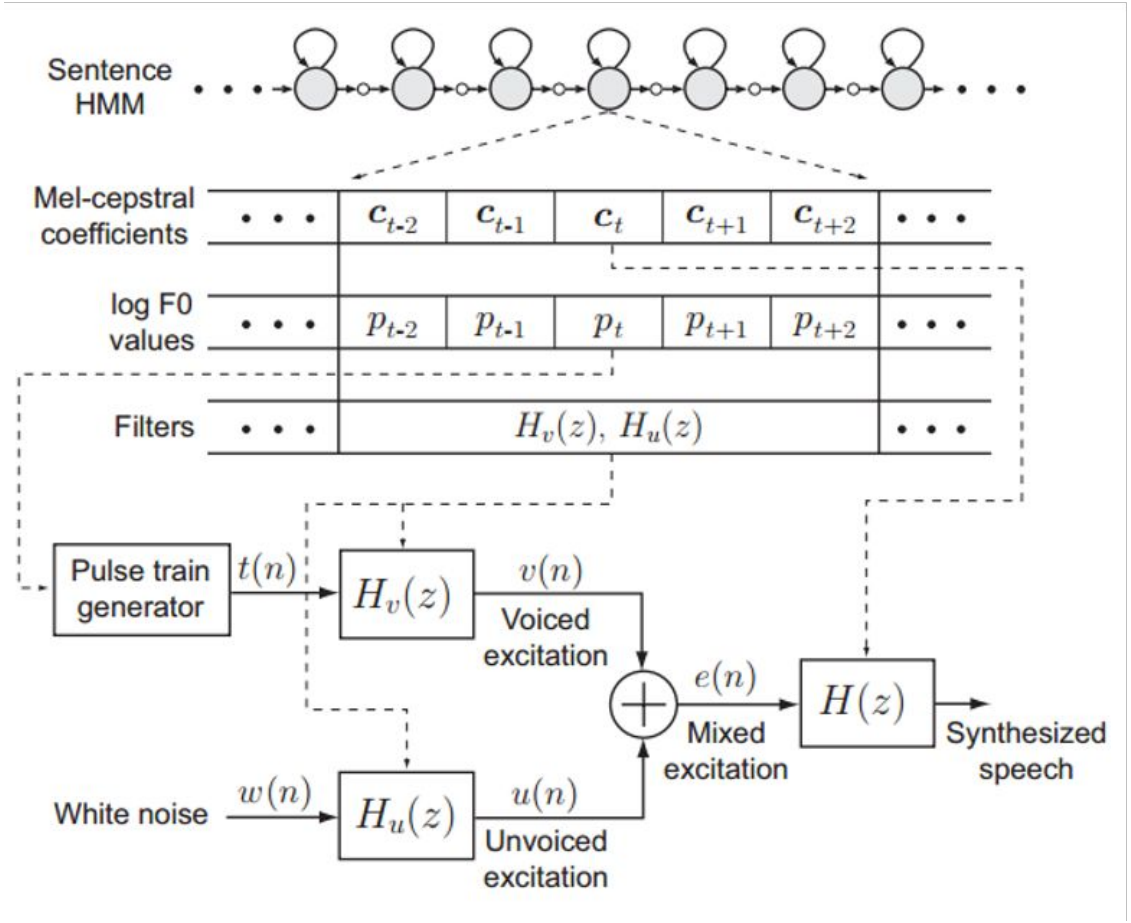
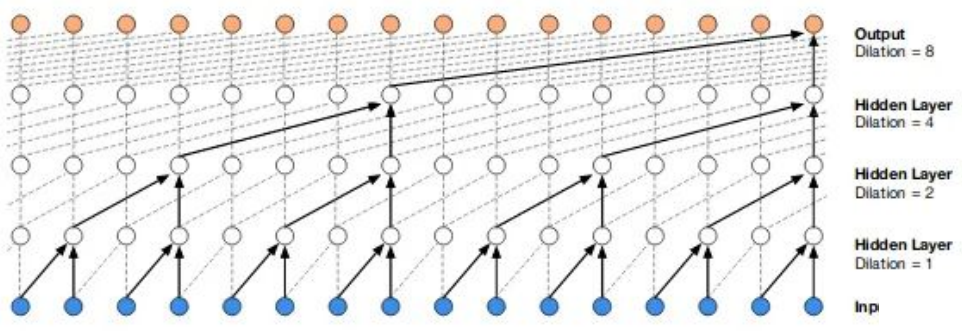


Figure: ML-based excitation scheme proposed by Maia et al. for HMM-based speech synthesis: filters Hu(z) and Hu(z) are associated with each state.

(Tokuda, Zen, & Black. 2009)

Learning neural vocoders from data

Vocoder: acoustic features to waveform



WaveNet dilated convolution architecture to handle audio sample rates

Method	Subjective 5-scale MOS
16kHz, 8-bit μ-law, 25h data:	
LSTM-RNN parametric [27]	3.67 ± 0.098
HMM-driven concatenative [27]	3.86 ± 0.137
WaveNet [27]	4.21 ± 0.081
24kHz, 16-bit linear PCM, 65h data:	
HMM-driven concatenative	4.19 ± 0.097
Autoregressive WaveNet	4.41 ± 0.069
Distilled WaveNet	4.41 ± 0.078

Table 1: Comparison of WaveNet distillation with the autoregressive teacher WaveNet, unit-selection (concatenative), and previous results from [27]. MOS stands for Mean Opinion Score.

https://arxiv.org/pdf/1711.10433.pdf

Key Questions in Parametric Synthesis

- Listen to the “low level” buzzy quality characteristic of most parametric systems
- Listen to clarity/impact of plosives compared to concatenative example

Parametric

Unit Selection

Comparing Vocoder / Excitation Models

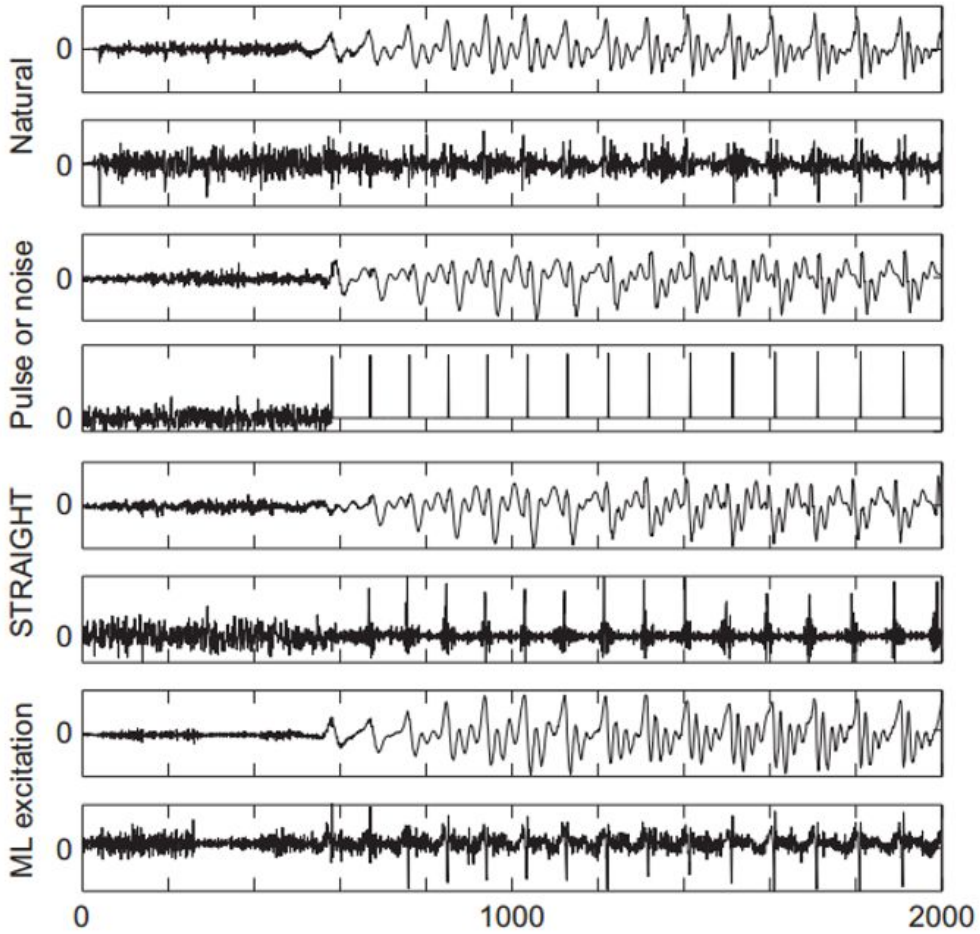


Figure: Waveforms from top to bottom: natural speech and its residual, speech and excitation synthesized with simple periodic pulse-train or white-noise excitation, speech and excitation synthesized with STRAIGHT vocoding method, and speech and excitation synthesized with ML excitation method.

(Tokuda, Zen, & Black. 2009)

Tacotron2 Architecture

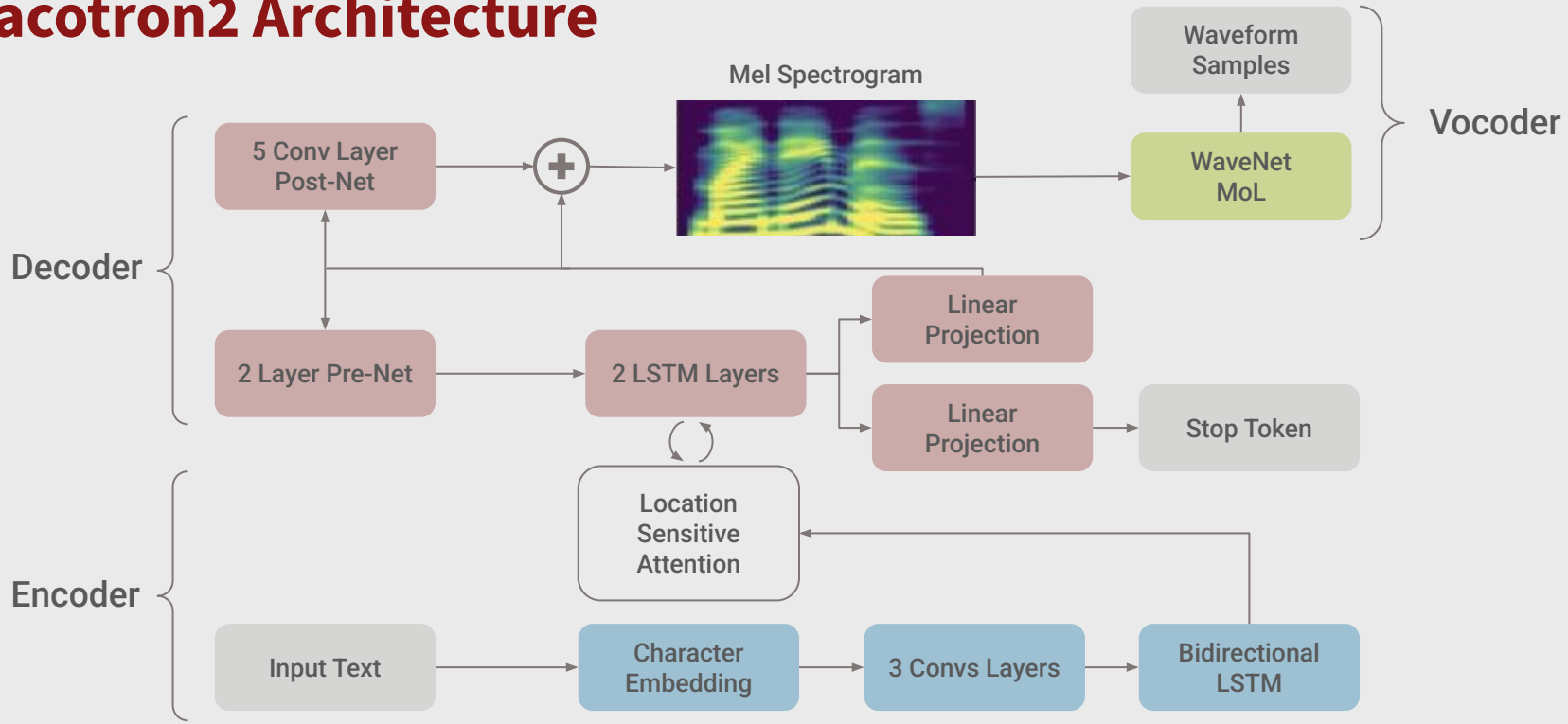


Figure: Tacotron2 architecture: An encoder-decoder maps from graphemes to mel spectrograms, followed by a vocoder that maps to wavefiles.Shen et al. (2018)

Conclusions

- **Common recipe for any TTS effort to achieve best results (data + evaluation are critical)**
- **Concatenative systems**
  - Easier to get working quickly (no modeling work)
  - Low level signal processing and joins cause artifacts – Ceiling on quality.
  - Require large single-speaker training sets for best coverage
- **”Editing prosody” is critical for human-like TTS and modern applications**
  - Parametric systems expose interfaces to predict/control duration, F0, etc.
  - No natural way to do this in concatenative systems
  - Parametric models have representation choices which impact TTS quality
  - Need prosody annotations in training data to create prosody controls
- **Up next: TTS with modern deep learning**

Thank You