CS 224S / Linguist 285
Spoken Language Processing

Andrew Maas | Stanford University | Spring 2024

Lecture 4: TTS Waveform Synthesis

Outline

- Recipe for building great TTS
- Concatenative Waveform Synthesis:
  - Diphone Synthesis
  - Unit Selection Synthesis
  - Joining Units
- What to Predict in Parametric Synthesis

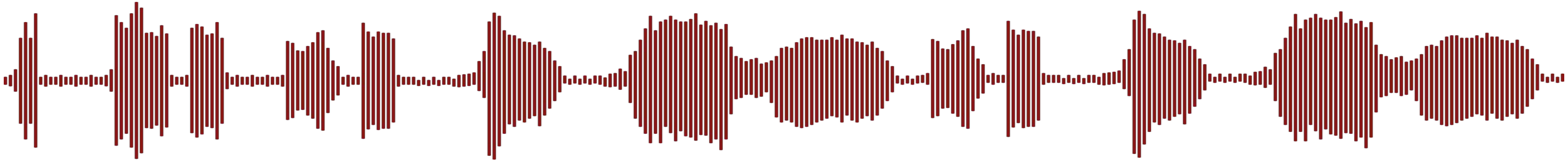
Recipe for Building Great TTS

The Two Stages of TTS

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

1. Text Analysis: Text into intermediate representation:

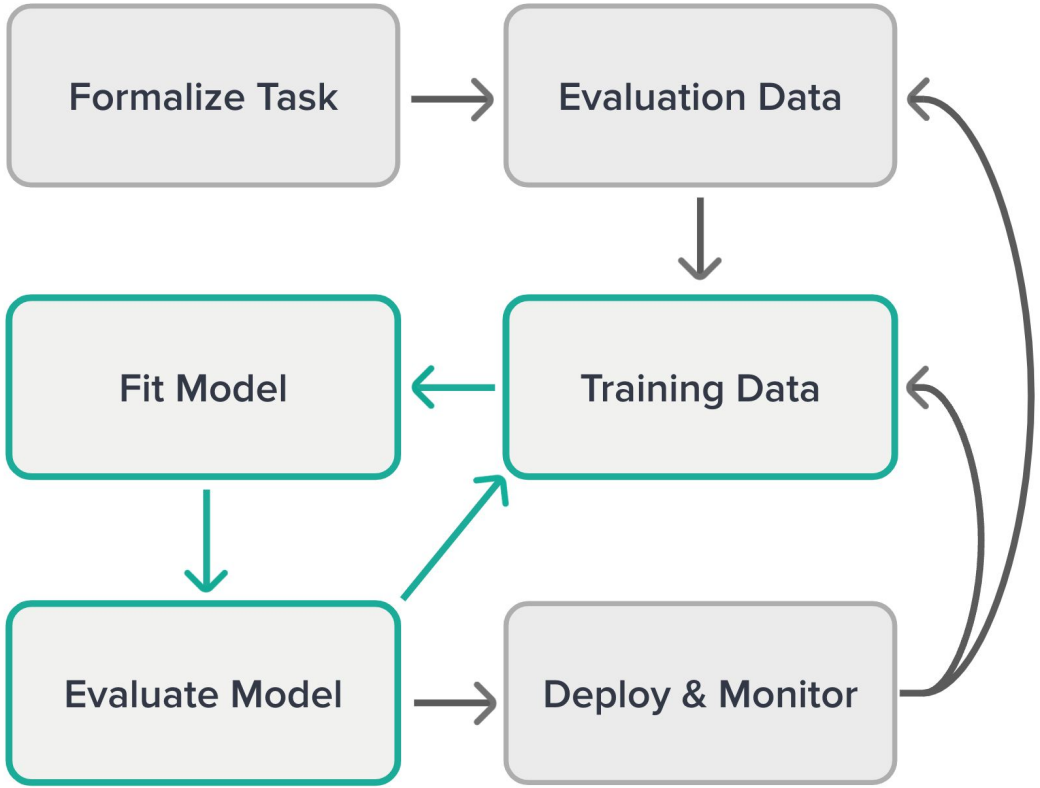
2. Waveform Synthesis: From the intermediate representation into waveform



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

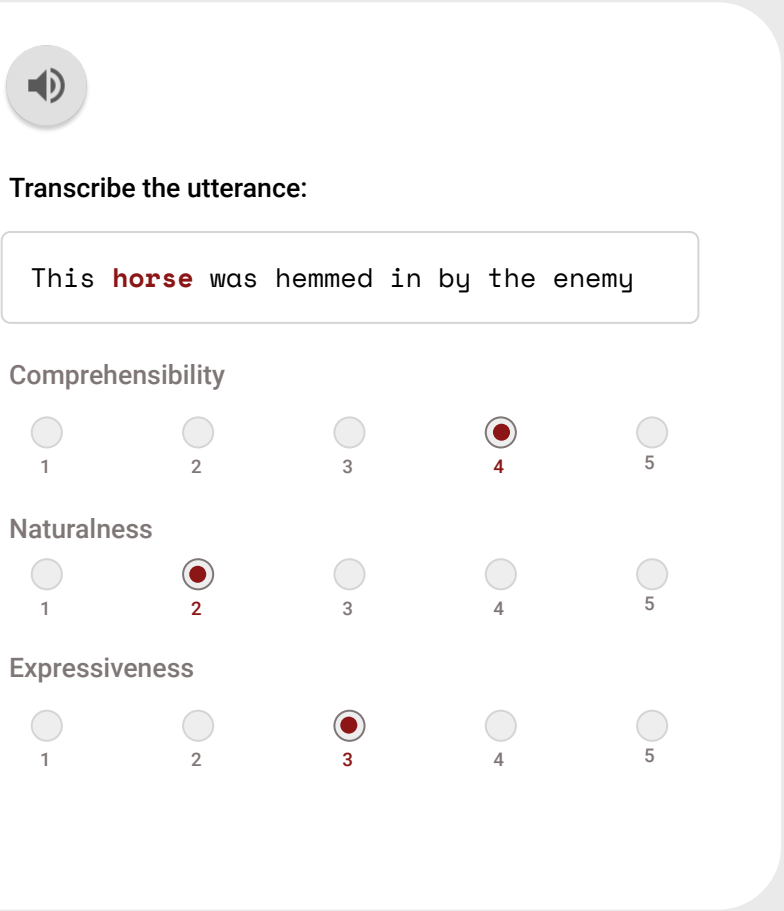
Evaluation of TTS

- **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:

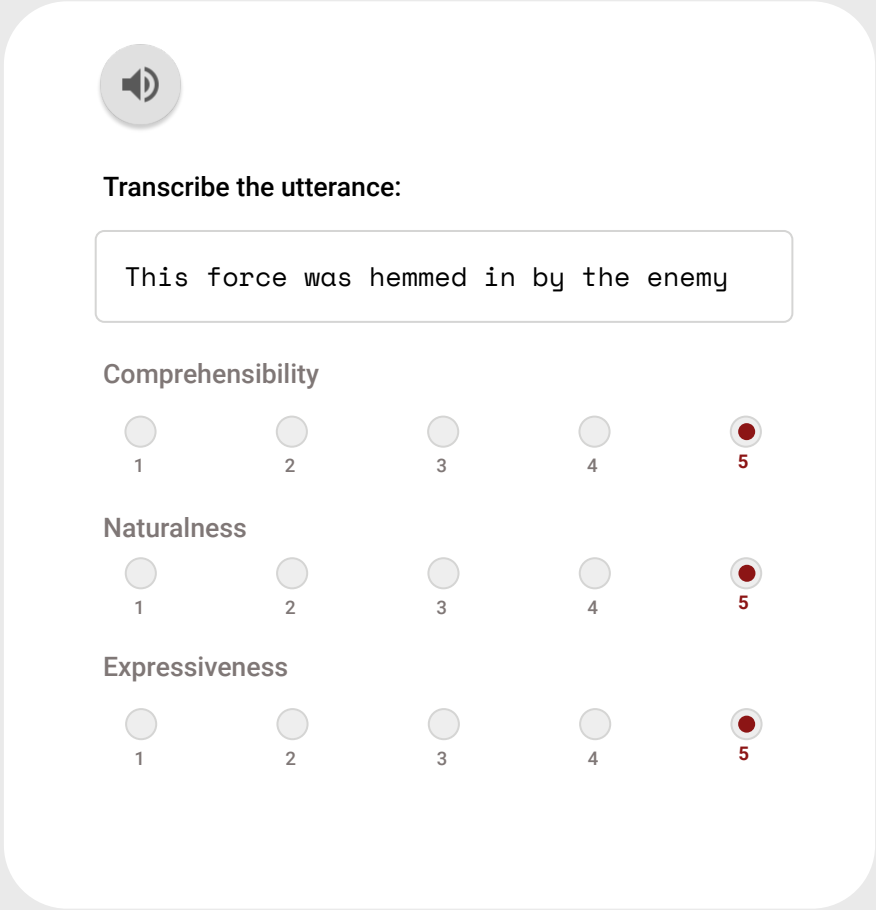
- **Intelligibility**
  - Usually quantified objectively via transcription
- **Comprehensibility**
  - How easy is it to understand a particular utterance
- **Naturalness**
  - How natural does the utterance sound
- **Expressiveness**
  - How well does the intonation match the substance of the utterance



Mean Opinion Score

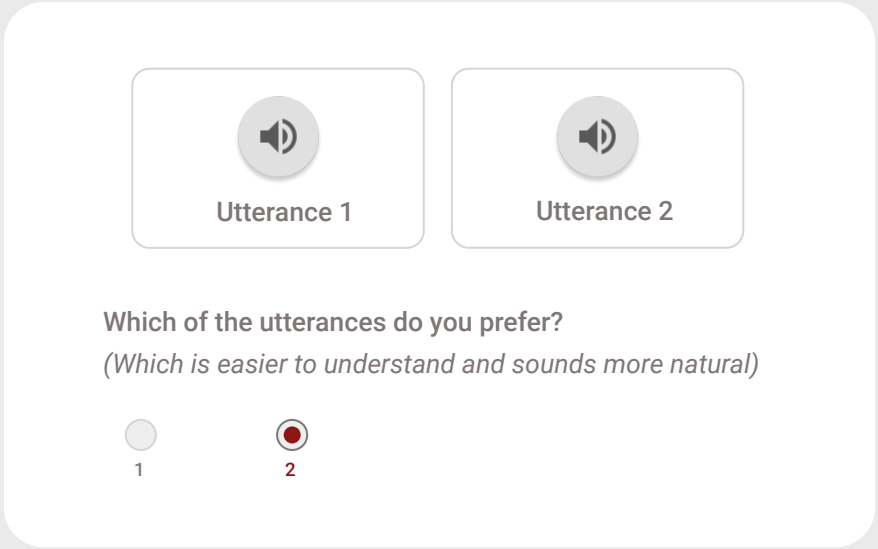
Using crowdsourced ranking of synthesis results based on:

- **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

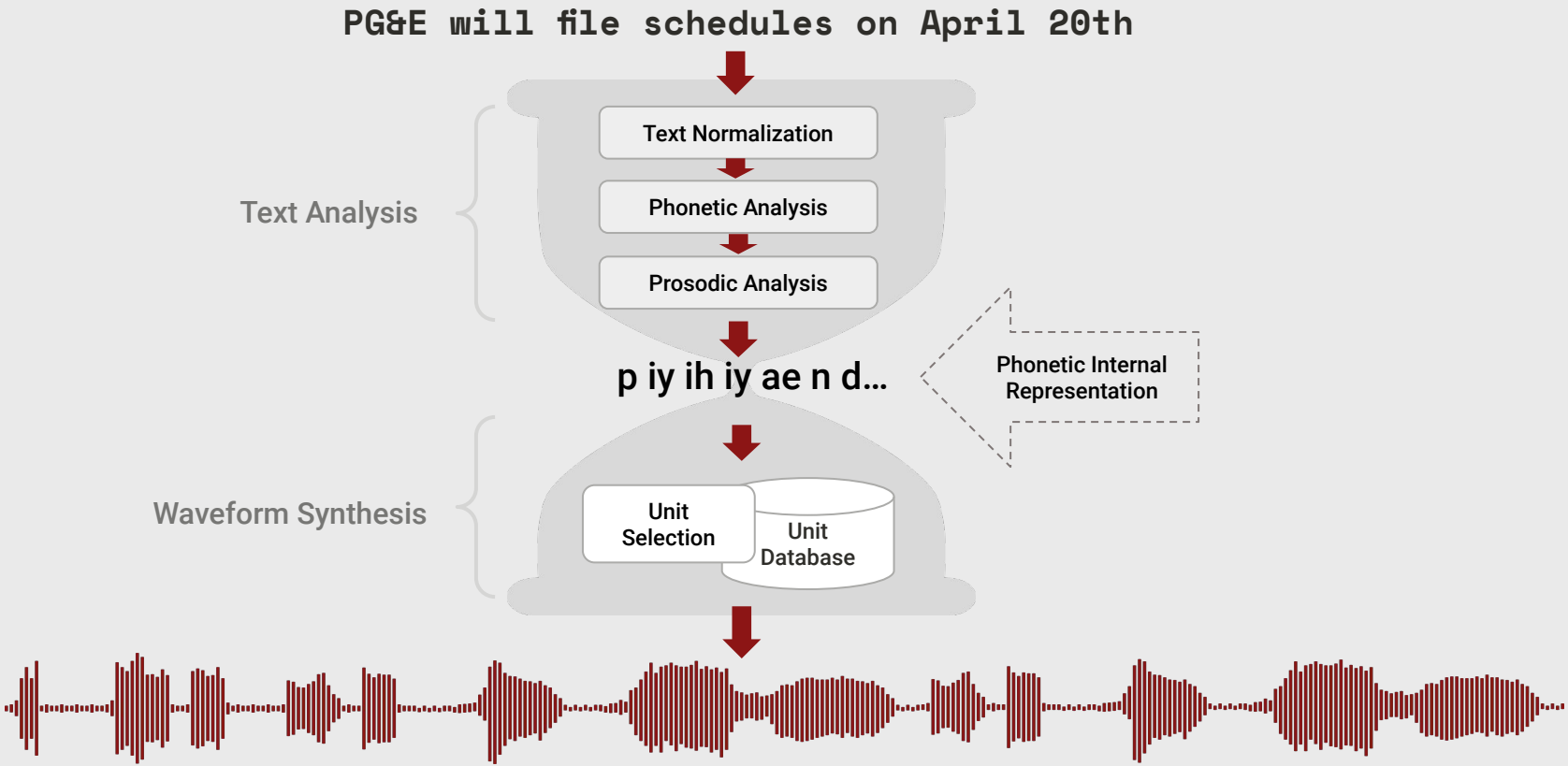
Using crowdsourced selections to elicit direct preferences for TTS settings



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

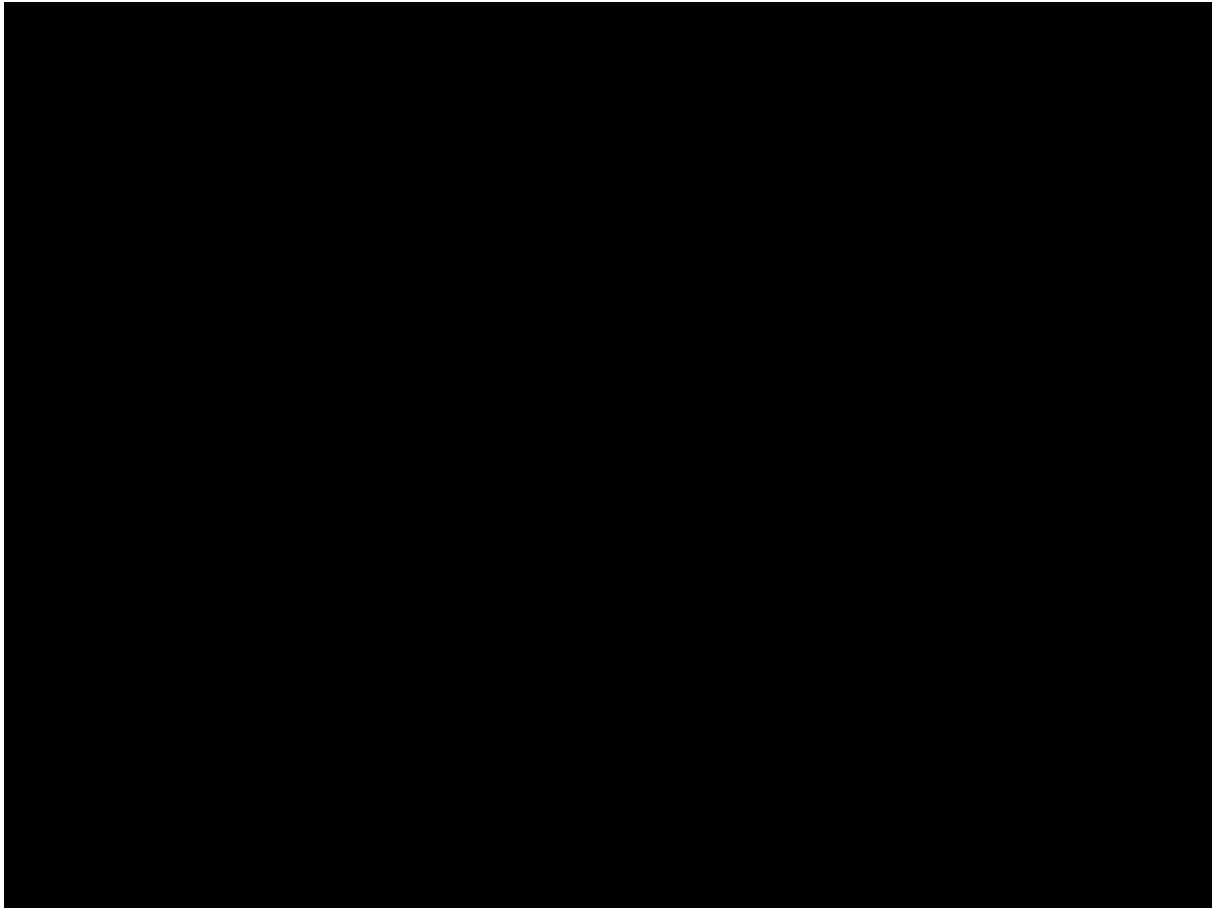
- **Great acoustic quality**
  - At least 16 kHZ
  - Good microphone
  - Minimal background noise (including page turning and breathing!)
- **Emotional and phonetic range to match application**
  - System will clone accent of single speaker
  - Must collect emotional speech if TTS needs to produce it
  - 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

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



https://drive.google.com/file/d/1vayhixbgUypP3glCN_xmYJEghh4PcAng/view?resourcekey

Tacotron2 Architecture

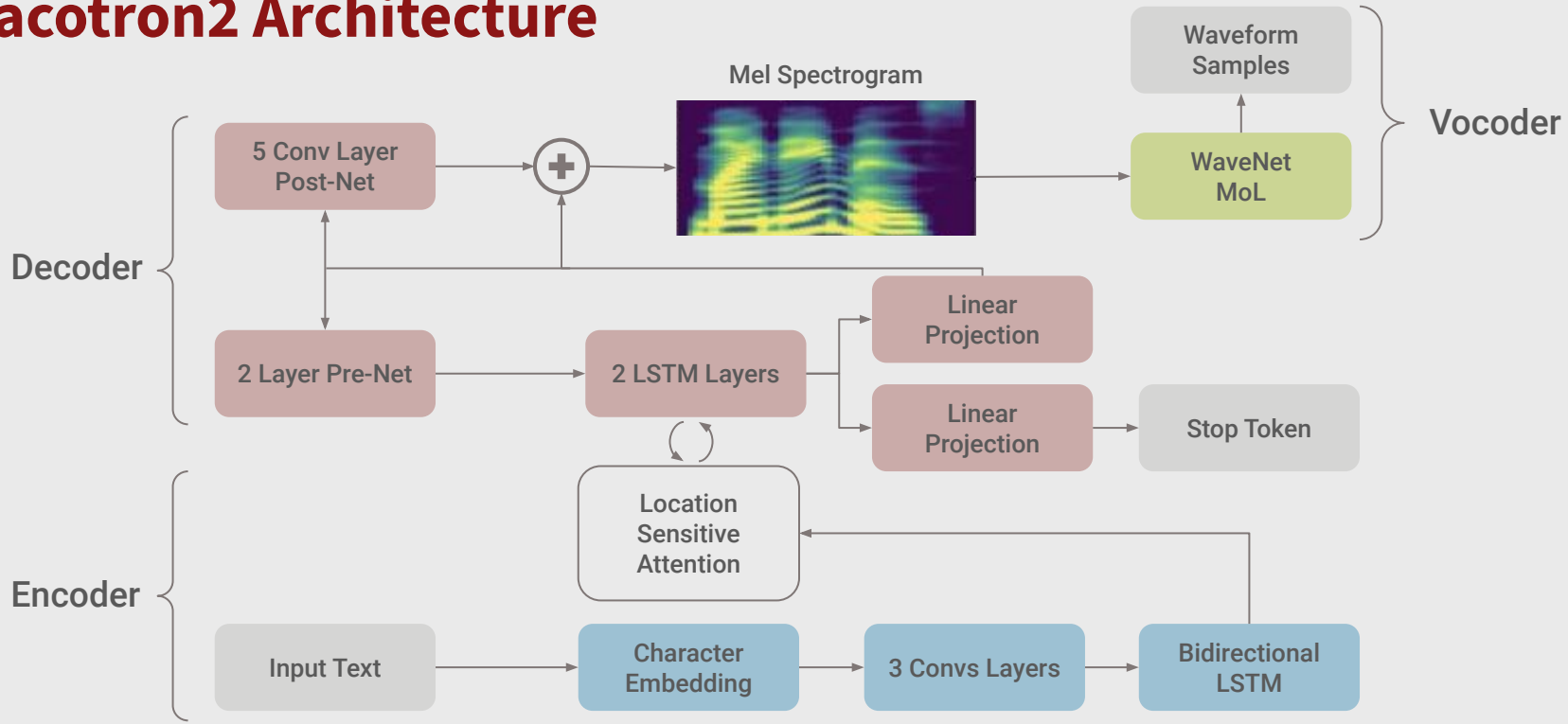


Figure: Tacotron2 architecture: An encoder-decoder maps from graphemes to mel spectrograms, followed by a vocoder that maps to wavefiles.Shen et al. (2018)

Concatenative Waveform Synthesis

- **Key steps::**
- **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

Waveform Synthesis

Given:

- **String of phones**
- **Prosody**
  - 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

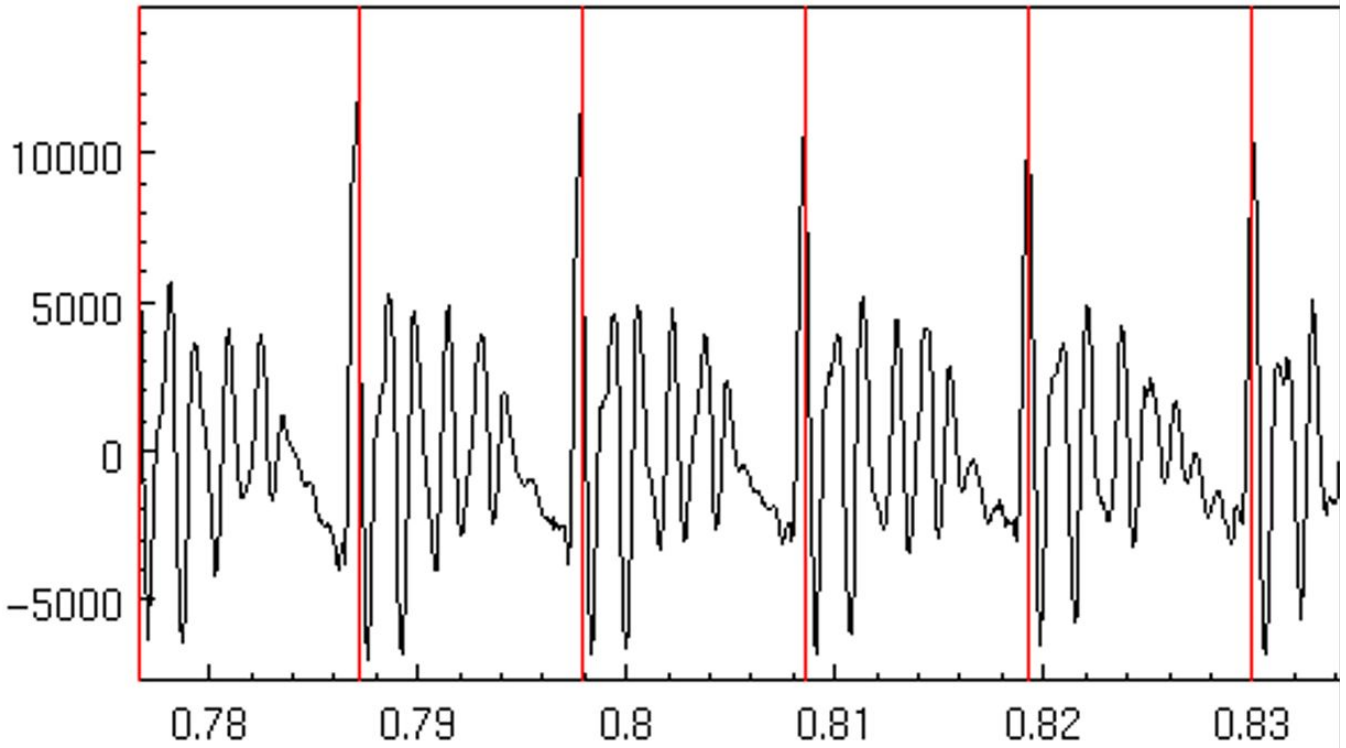
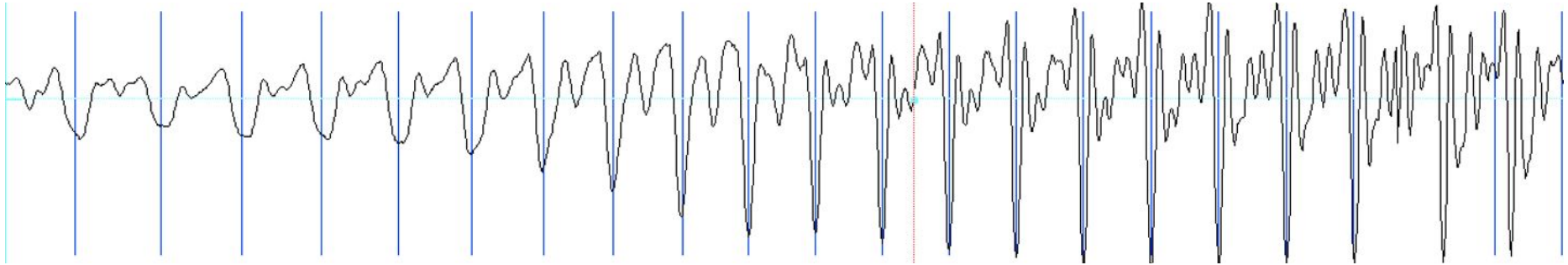


Figure: Alan Black

Epoch-labeling

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



Duration Modification

- Duplicate/remove short term signals

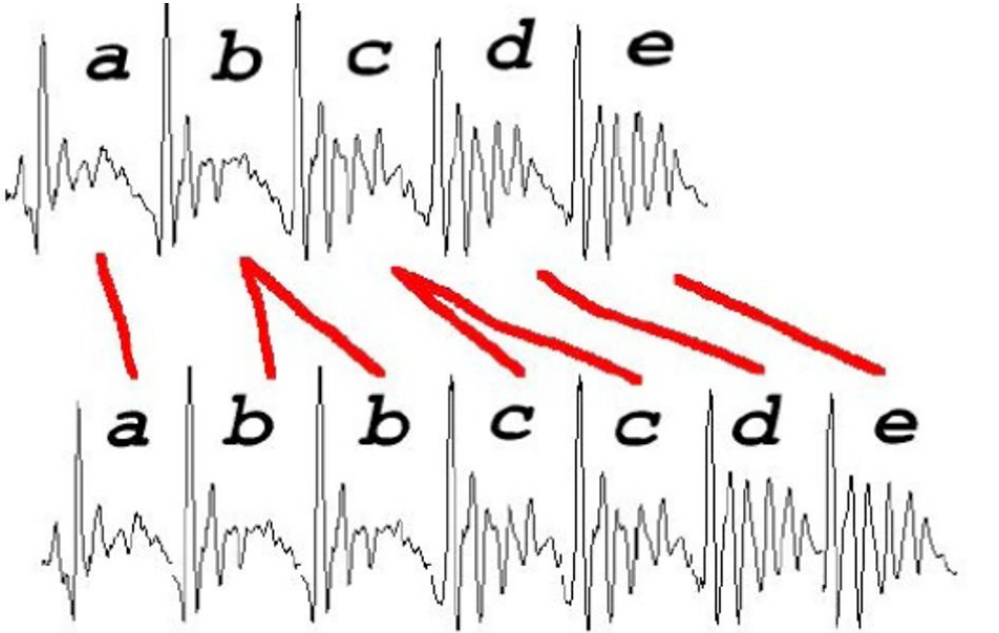


Figure: Richard Sproat

Pitch Modification

- Move short-term signals closer together/further apart

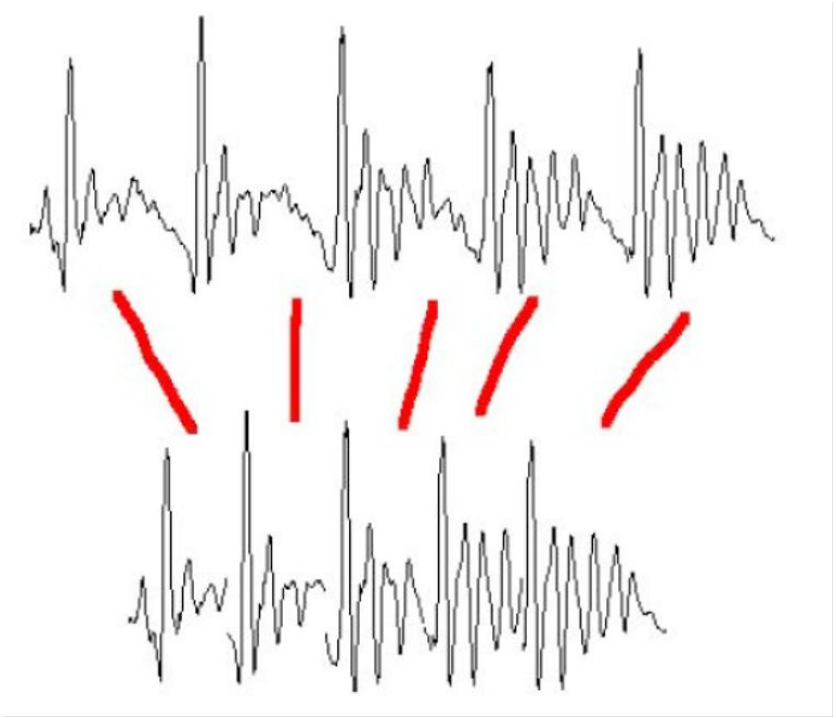


Figure: Richard Sproat

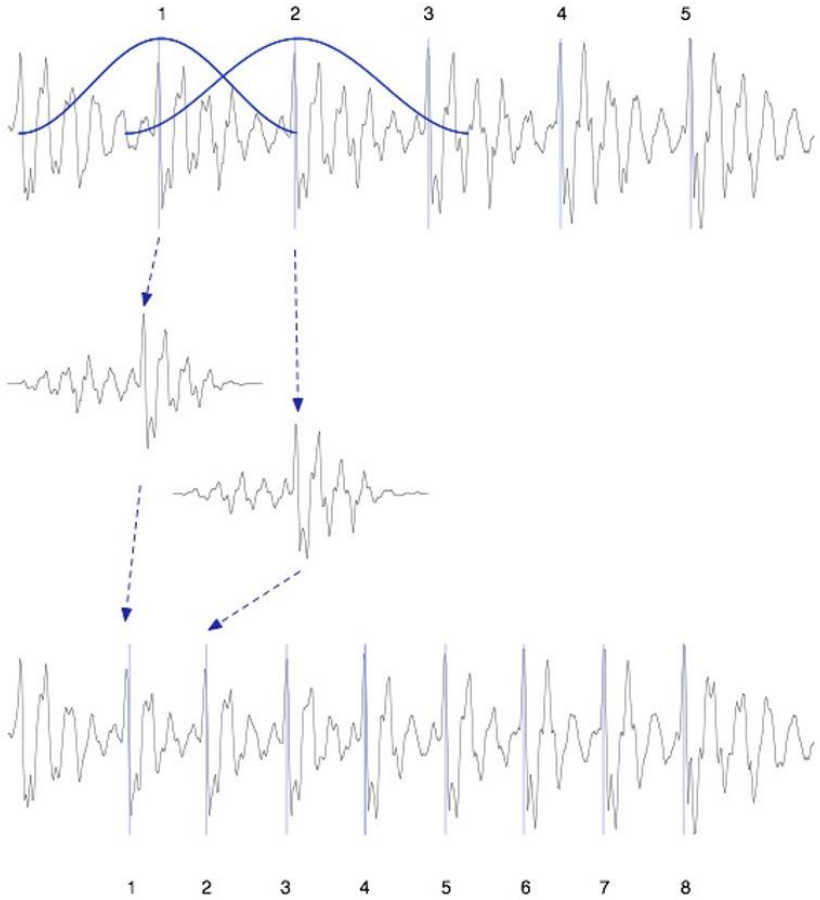
TD-PSOLA ™

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?

What Does the HMM* Produce?

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

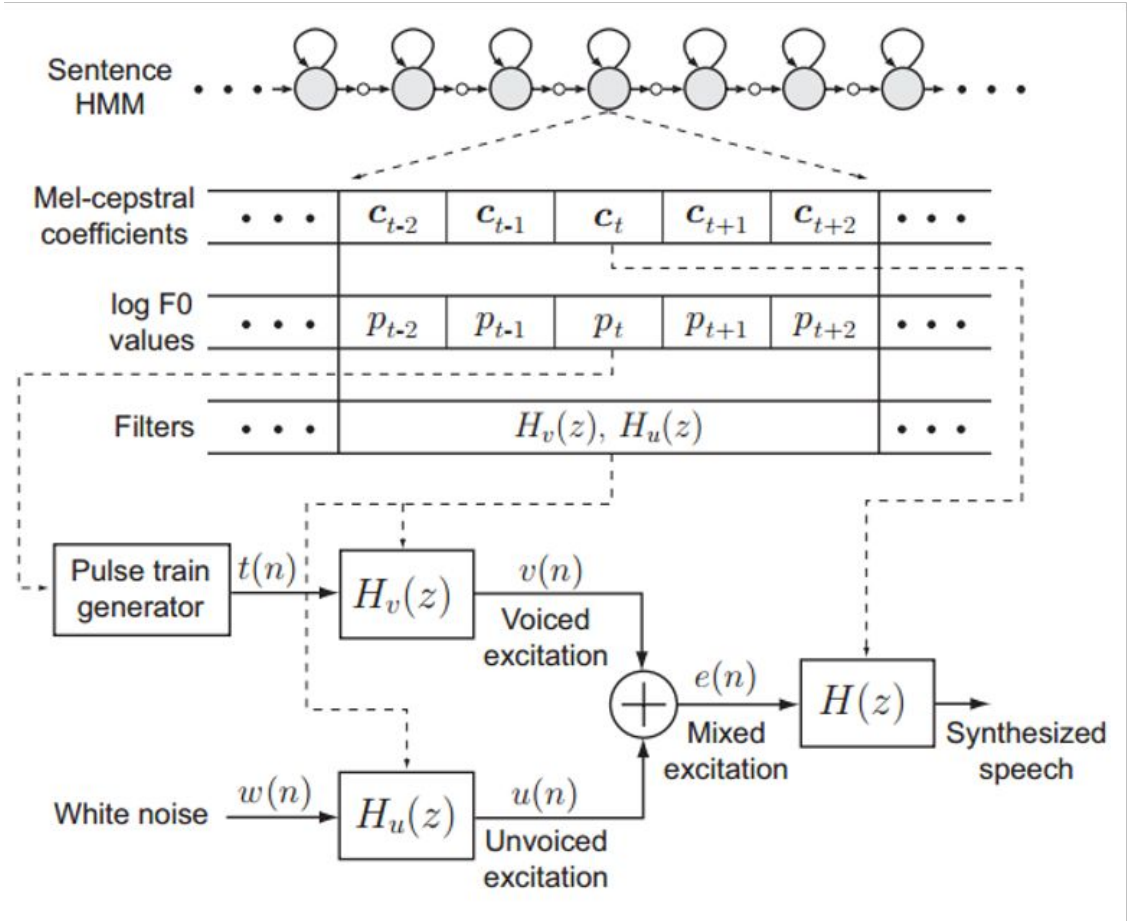
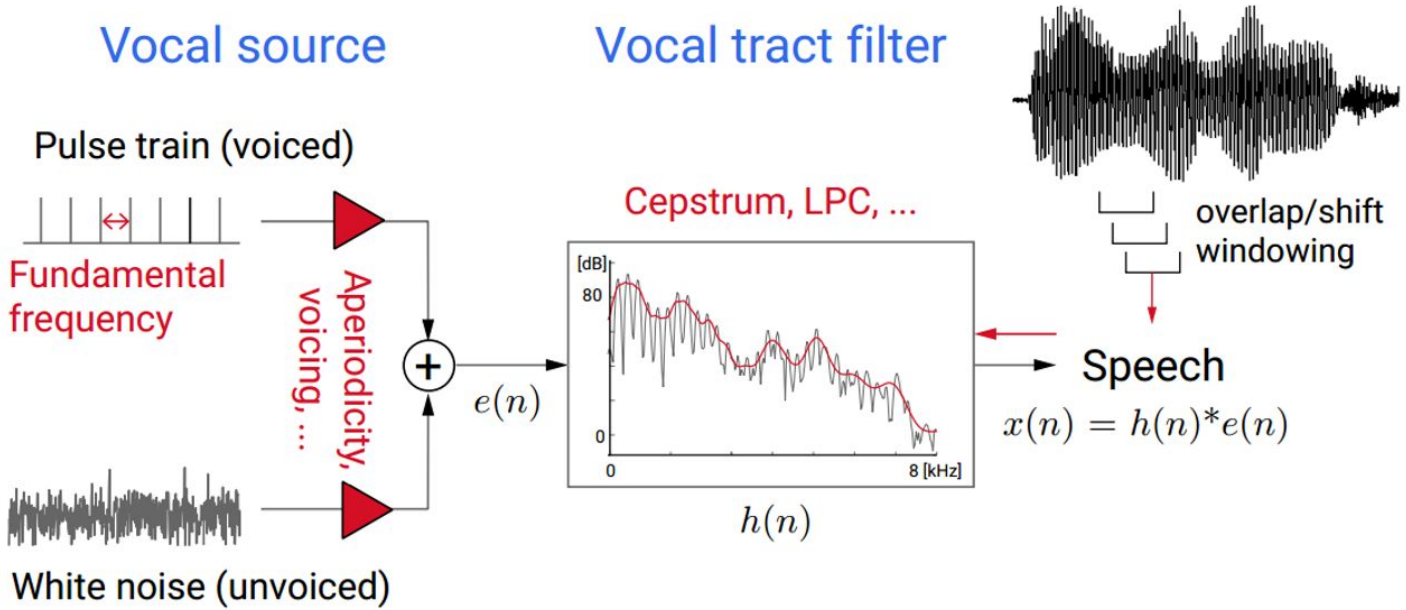


Figure: ML-based excitation scheme proposed by Maia et al. for HMM-based speech synthesis: filters Hu(z) and Hu(z) are associated with each state.

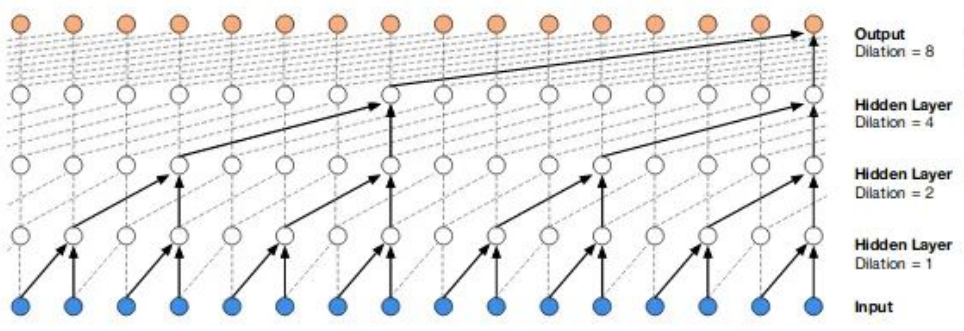
(Tokuda, Zen, & Black. 2009)

Synthesis with Source-filter Model

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



Learning neural vocoders from data



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

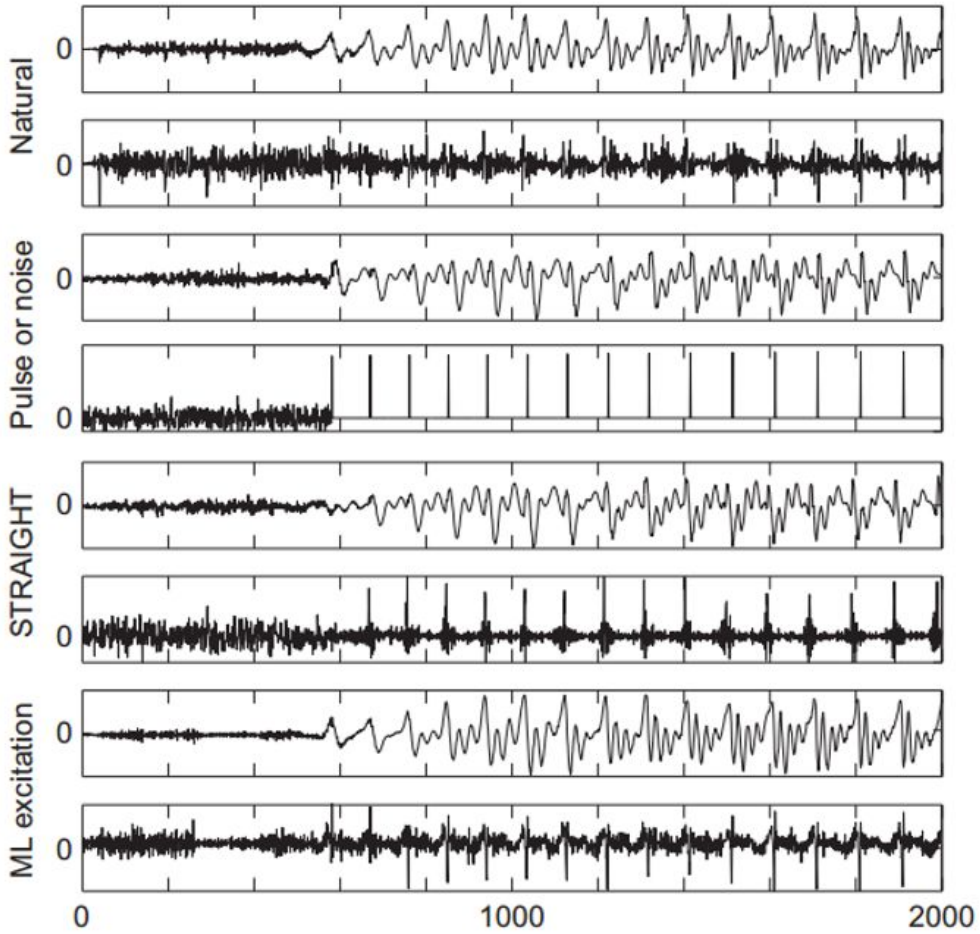


Figure: Waveforms from top to bottom: natural speech and its residual, speech and excitation synthesized with simple periodic pulse-train or white-noise excitation, speech and excitation synthesized with STRAIGHT vocoding method, and speech and excitation synthesized with ML excitation method.

(Tokuda, Zen, & Black. 2009)

Tacotron2 Architecture

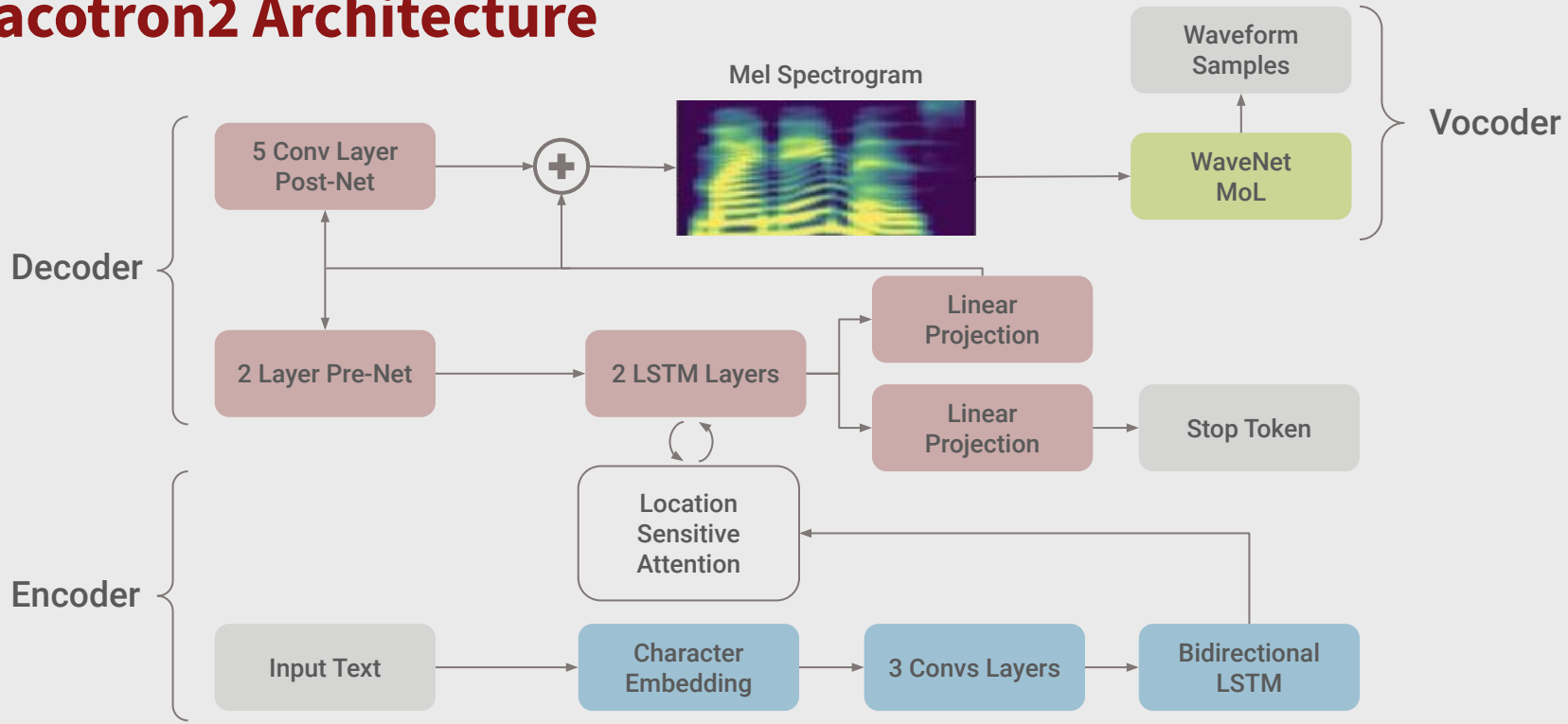


Figure: Tacotron2 architecture: An encoder-decoder maps from graphemes to mel spectrograms, followed by a vocoder that maps to wavefiles.Shen et al. (2018)

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

Appendix

Building Diphone Schemata

- **Find list of phones in language:**
  - Plus interesting allophones
  - Stress, tons, clusters, onset/coda, etc
  - Foreign (rare) phones
- **Build carriers for:**
  - Consonant-vowel, vowel-consonant
  - Vowel-vowel, consonant-consonant
  - Silence-phone, phone-silence
  - Other special cases
- **Check the output:**
  - List all diphones and justify missing ones
  - Every diphone list has mistakes

Slide: Richard Sproat

Recording Conditions

- **Ideal:**
  - Anechoic chamber
  - Studio quality recording
  - EGG signal
- **More likely:**
  - Quiet room
  - Cheap microphone/sound blaster
  - No EGG
  - Head mounted microphone
- **What we can do:**
  - Repeatable conditions
  - Careful setting on audio levels

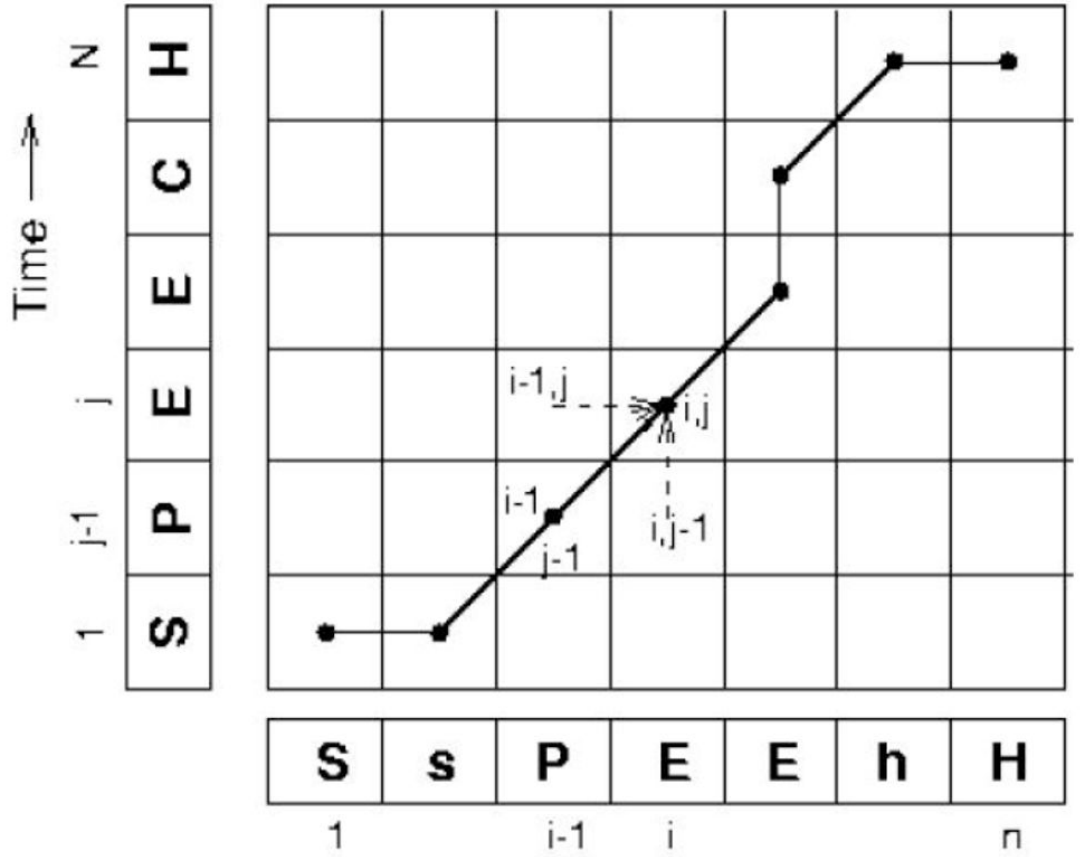
Slide: Richard Sproat

Labeling Diphones

- **Run a speech recognizer in forced alignment mode**
  - Forced alignment:
    - Given: A trained ASR system, a wav file, a transcriptions
    - Returns: an alignment of the phones to the wavfile
- **Much easier than phonetic labeling:**
  - Words and phone sequence are defined
  - They are clearly articulated
  - But sometimes speaker still pronounces wrong, so need to check
- **Phone boundaries less important**
  - +- 10 ms is okay
- **Midphone boundaries important**
  - Where is the stable part
  - Can it be automatically found?

Slide: Richard Sproat

Dynamic Time Warping



Concatenating Diphones: Junctures

- **If waveforms are very different, will perceive a click at the junctures**
  - So need to window them
- **Also if both diphones are voiced**
  - Need to join them pitch-synchronously
- **That means we need to know where each pitch period begins, so we can paste at the same place in each pitch period**
  - Pitch marking or epoch detection: mark where each pitch pulse or epoch occurs
    - Finding the Instant of Glottal Closure (IGC)
    - (note difference from pitch tracking)

Epoch-labeling: Electroglottograph (EGG) = Laryngograph, Lx

- Straps on speaker's neck near larynx
- Sends small high frequency current through adam's apple
- Human tissue conducts well; air not as well
- Transducer detects how open the glottis is (I.e. amount of air between folds) by measuring impedance.

Image: UCLA Phonetics Lab

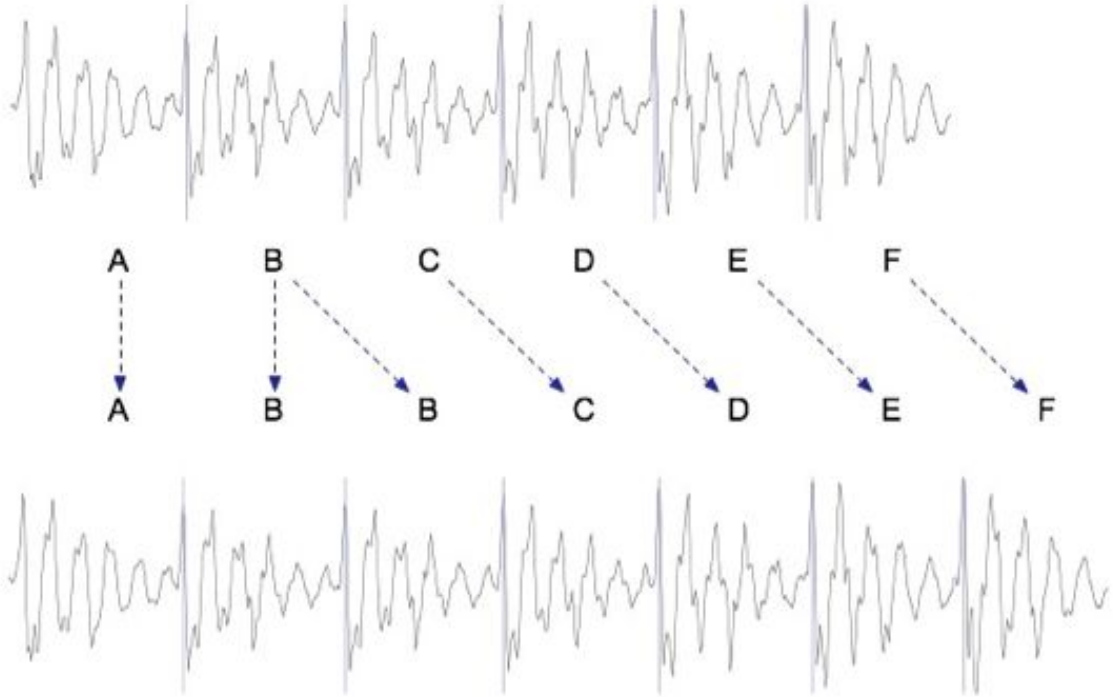
Prosodic Modification

- Modifying pitch and duration independently
- Changing sample rate modifies both:
  - Chipmunk speech
- Duration: duplicate/remove parts of the signal
- Pitch: resample to change pitch

Slide: Alan Black

Duration Modification

- Duplicate/remove short term signals



Overlap-and-add (OLA)

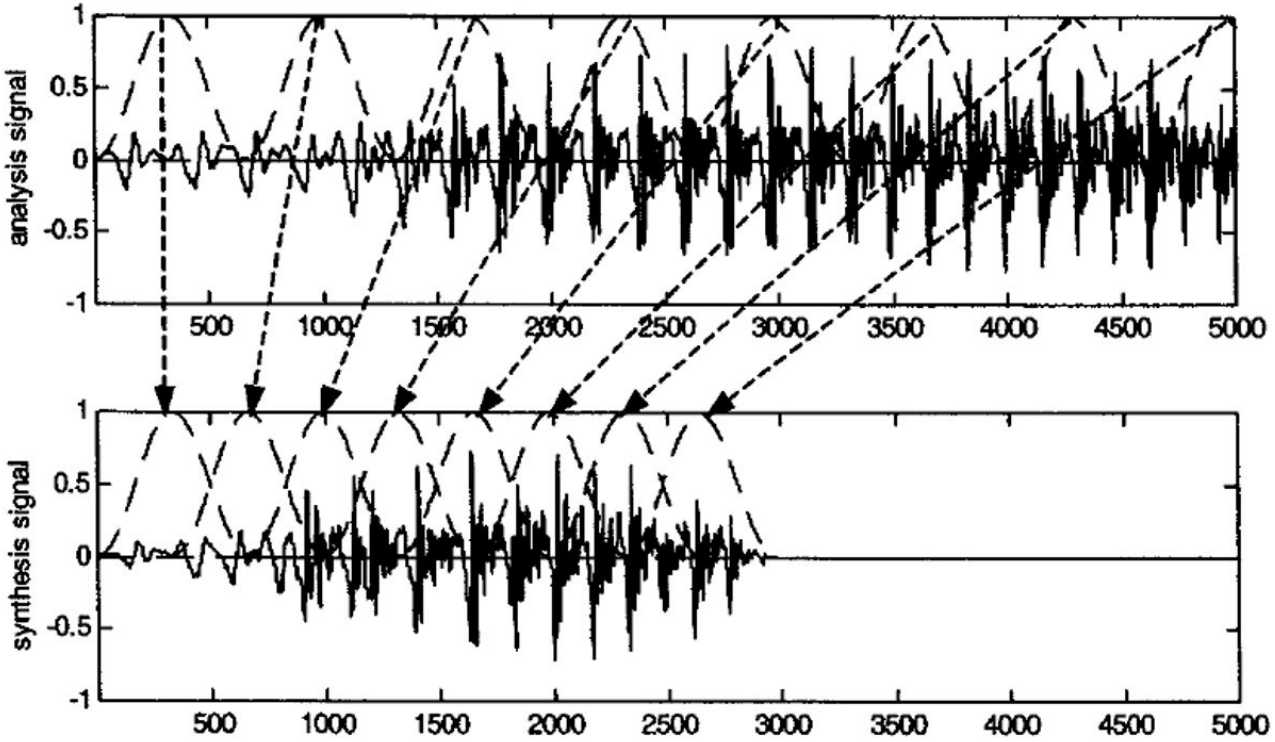


Image: Huang, Acero and Hon

Windowing

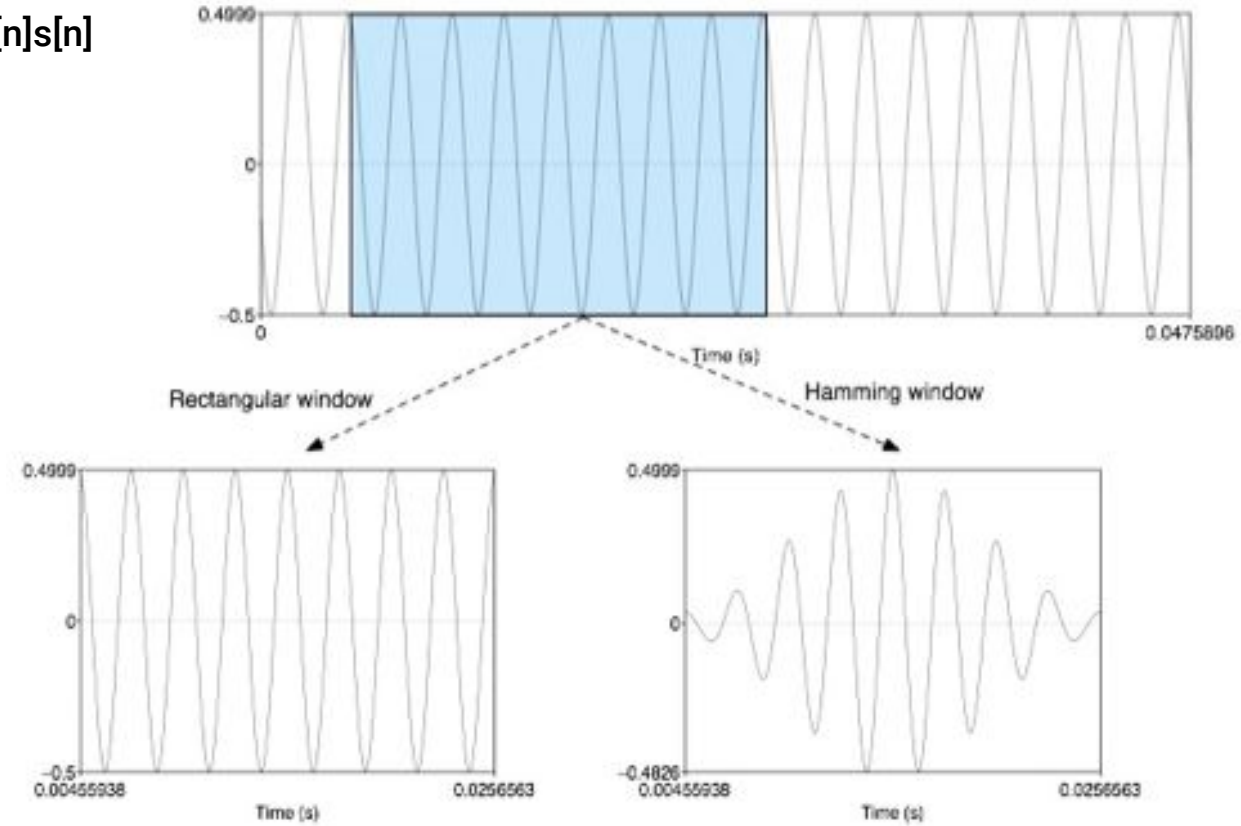
- Multiply value of signal at sample number n by the value of a windowing function
- y[n] = w[n]s[n]

$$rectangular \qquad w[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

$$hamming \qquad w[n] = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{L}) & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

Windowing

- y[n] = w[n]s[n]



Overlap and Add (OLA)

- Hanning windows of length 2N used to multiply the analysis signal
- Resulting windowed signals are added
- Analysis windows, spaced 2N
- Synthesis windows, spaced N
- Time compression is uniform with factor of 2
- Pitch periodicity somewhat lost around 4th window

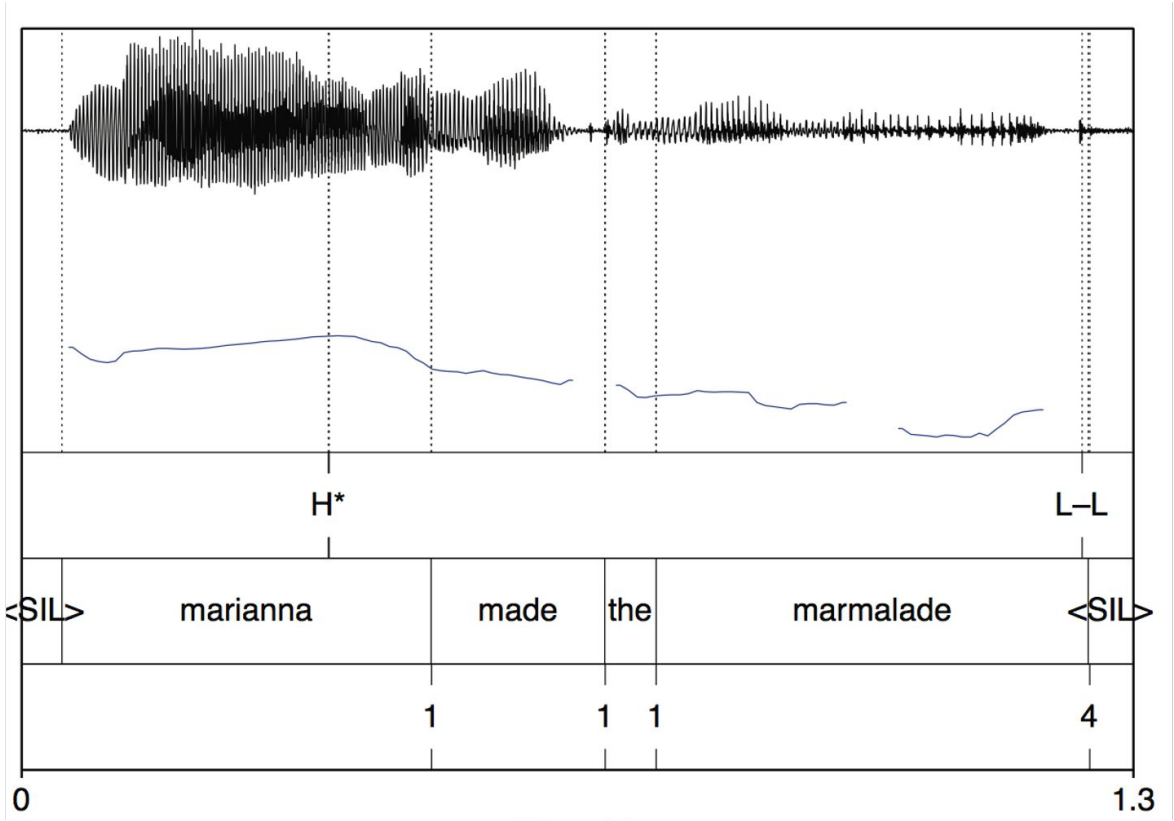
Predicting Intonation in TTS

- **Prominence/Accent:** Decide which words are accented, which syllable has accent, what sort of accent
- **Boundaries:** Decide where intonational boundaries are
- **Duration:** Specify length of each segment
- **F0:** Generate F0 contour from these

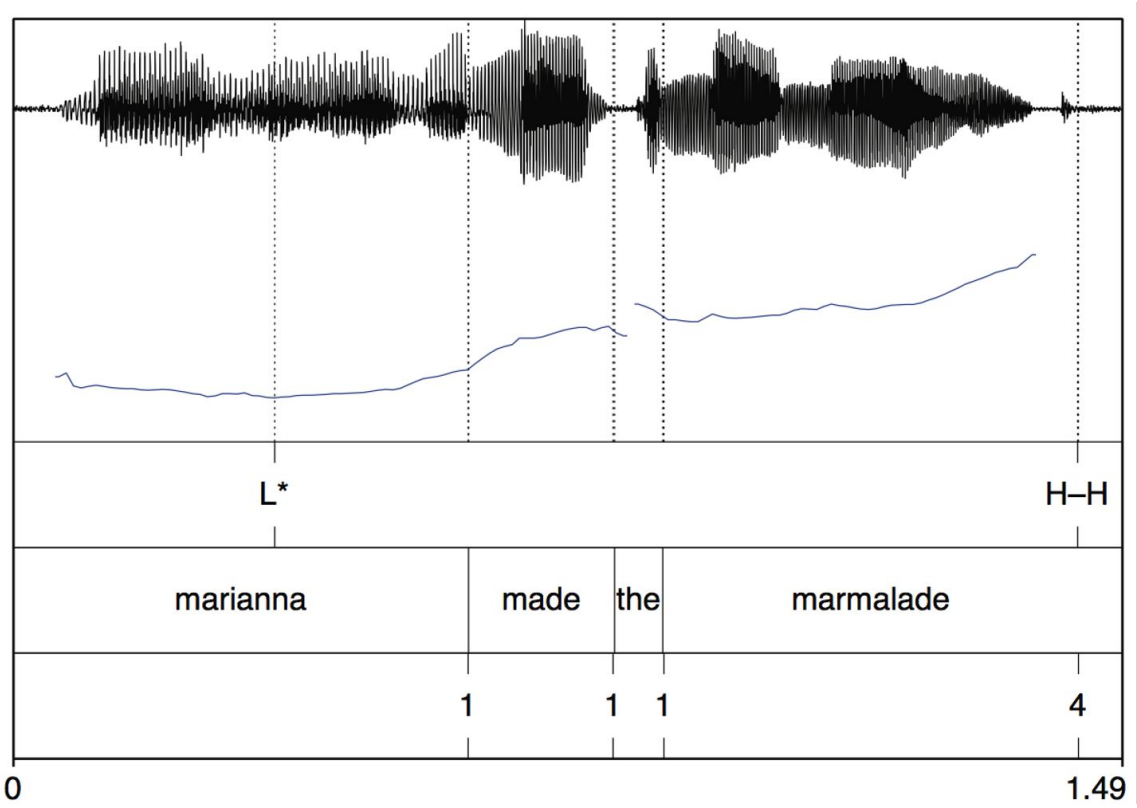
ToBI: Tones and Break Indices

- **Pitch accent tones**
  - H* “peak accent”
  - L* “low accent”
  - L+H* “rising peak accent” (contrastive)
  - L*+H ‘scooped accent’
  - H+!H* downstepped high
- **Boundary tones**
  - L-L% (final low; Am English Declarative contour)
  - L-H% (continuation rise)
  - H-H% (yes-no question)
- **Break indices**
  - 0: clitics, 1, word boundaries, 2 short pause
  - 3 intermediate intonation phrase
  - 4 full intonation phrase/final boundary

ToBI: Tones and Break Indices



ToBI: Tones and Break Indices



Examples of the TOBI system

```
I don't eat beef
   L*     L* L*L-L%
```

Examples of the TOBI system

Marianna made the marmalade

H* L-L%

Marianna made the marmalade

L* H-H%

Examples of the TOBI system

"I" means insert

H* H* H*L-L%

1

"I" means insert

H*L- H*L-L%

3

ToBI

- Silverman, K., Beckman, M., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., and Hirschberg, J. (1992). ToBI: a standard for labelling English prosody. In Proceedings of ICSLP92, volume 2, pages 867-870
- Pitrelli, J. F., Beckman, M. E., and Hirschberg, J. (1994). Evaluation of prosodic transcription labeling reliability in the ToBI framework. In ICSLP94, volume 1, pages 123-126
- Pierrehumbert, J., and J. Hirschberg (1990) The meaning of intonation contours in the interpretation of discourse. In P. R. Cohen, J.Morgan, and M. E. Pollack, eds., Plans and Intentions in Communication and Discourse, 271-311. MIT Press.
- Beckman and Elam. Guidelines for ToBI Labelling. Web.

Generating the Mean of Each State

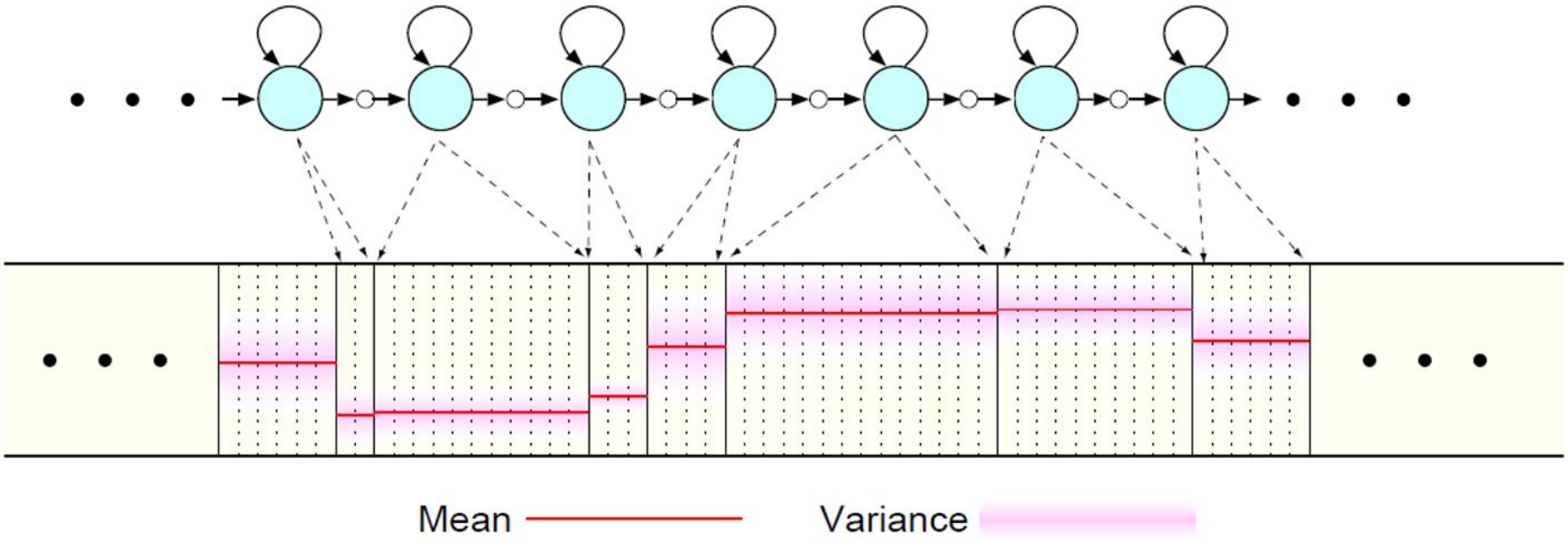
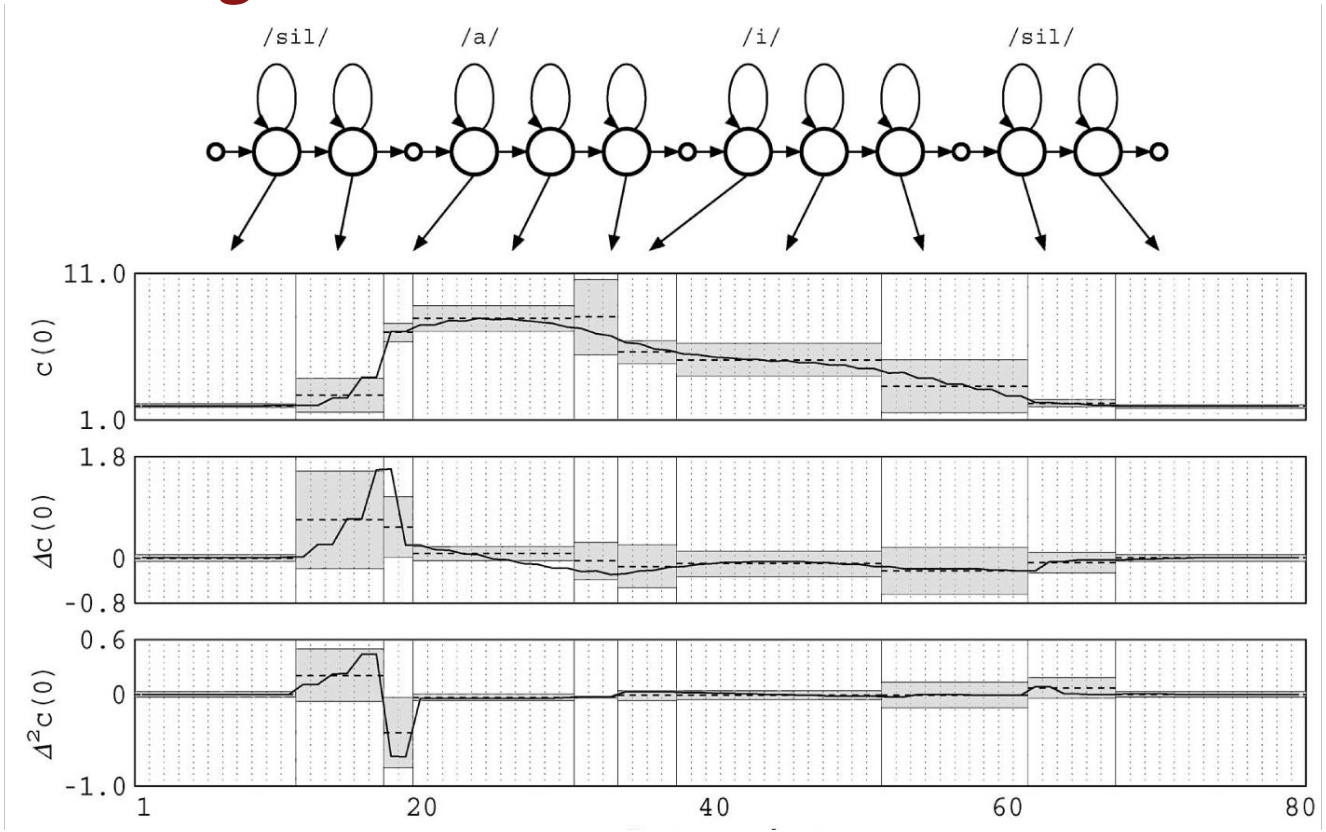


Figure: Tokuda and Zen 2009

Observations generated from HMM



Choosing a Sequence of Means Constrained by Deltas and Double-Deltas

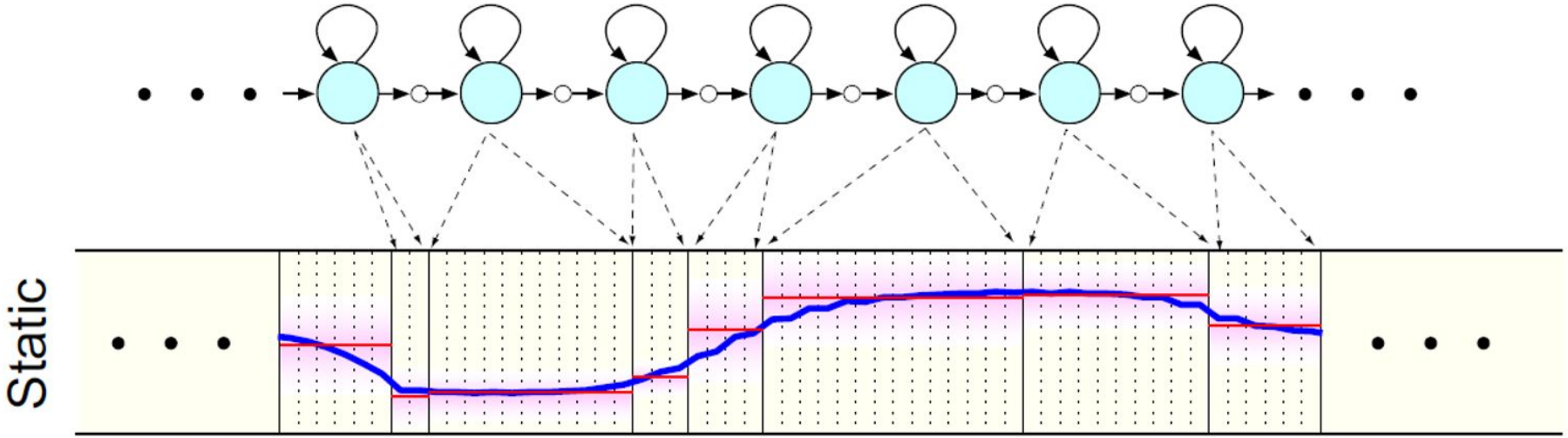
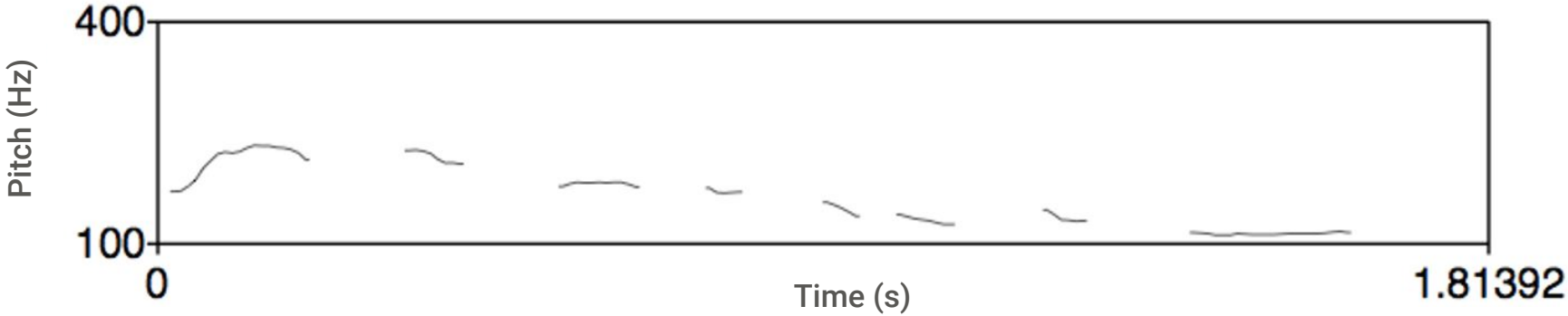


Figure: Tokuda and Zen 2009

Declination

- F0 tends to decline throughout a sentence



F0 Generation by Rule

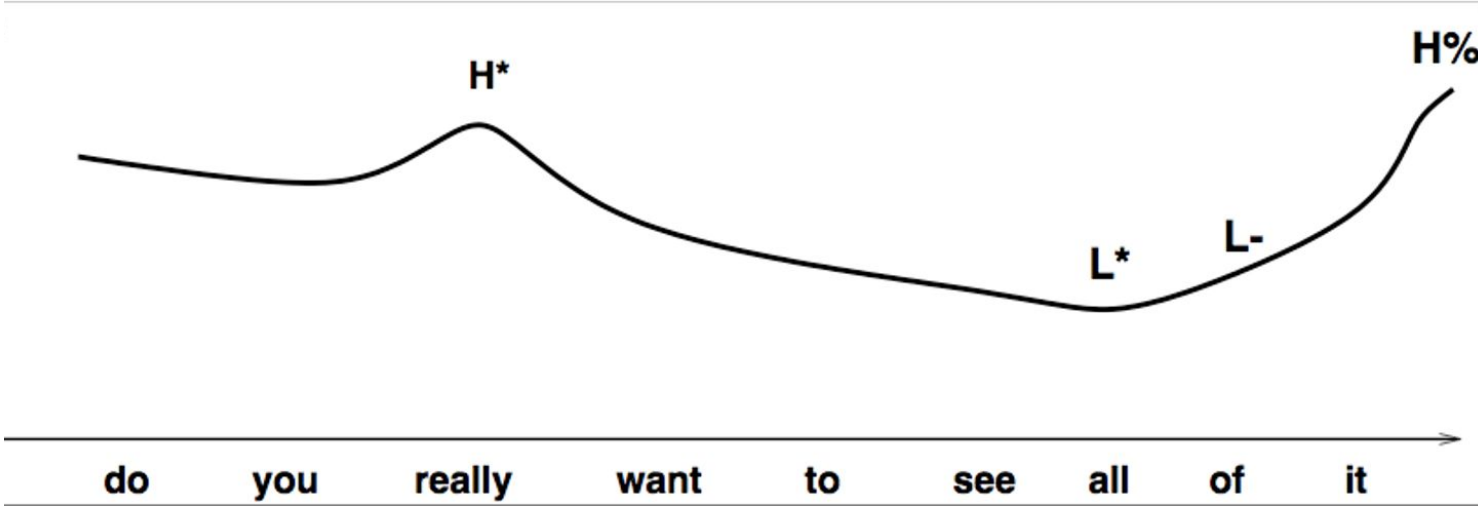
- **Generate a list of target F0 points for each syllable. For example:**
- **Generate simple H* “hat” accent (fixed speaker-specific F0 values) with 3 pitch points: [110, 140, 100]**
  - Modified by
  - gender,
  - declination,
  - end of sentence,
  - etc.

F0 Generation by Regression

- **Supervised machine learning**
- **Predict: value of F0 at 3 places in each syllable**
- **Predictor features:**
  - Accent of current word, next word, previous
  - Boundaries
  - Syllable type, phonetic information
  - Stress information
- **Need training sets with pitch accents labeled**
  - F0 is generally defined relative to pitch range
- **Range between baseline and topline frequency in an utterance**
- **Modern systems us ML to learn F0 generation**

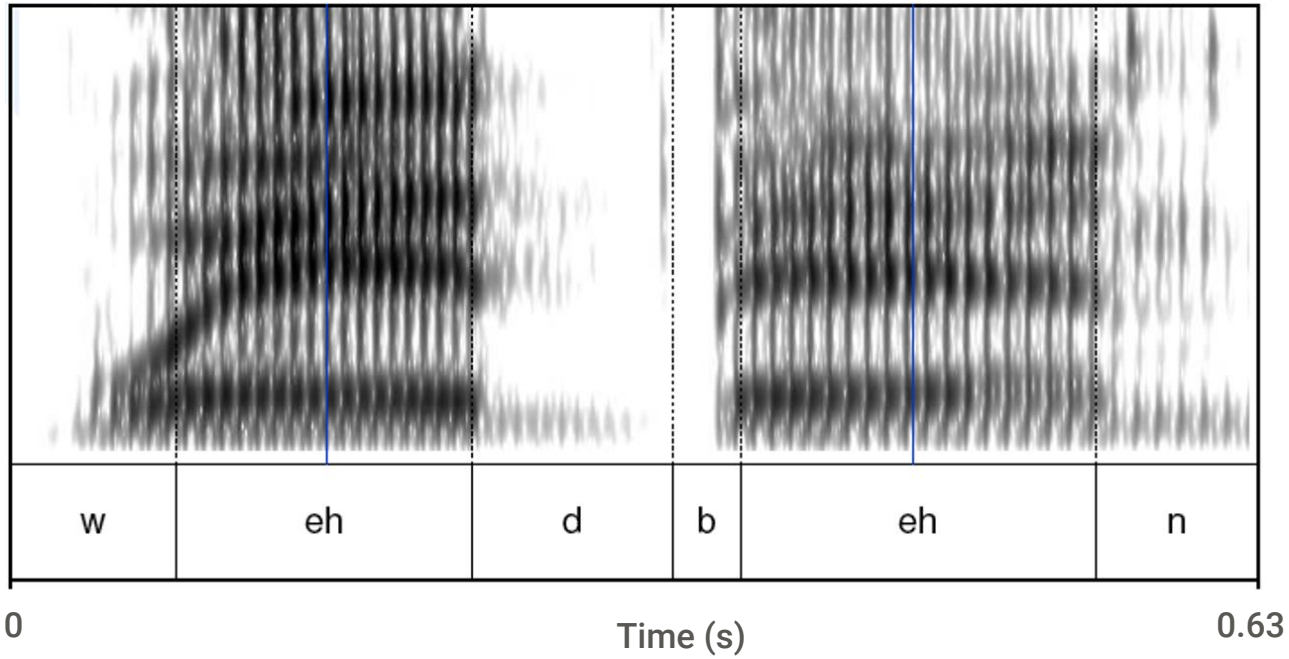
Internal Representation: Input to Waveform Synthesis

				H*										L*		L- H%			
do		you		really				want				to		see		all		of		it	
d	uw	y	uw	r	ih	l	iy	w	aa	n	t	t	ax	s	iy	ao	l	ah	v	ih	t
110	110	50	50	75	64	57	82	57	50	72	41	43	47	54	130	76	90	44	62	46	220



Diphones

- Mid-phone is more stable than edge:



Diphone TTS architecture

- **Training:**
  - Choose units (kinds of diphones)
  - Record 1 speaker saying 1 example of each diphone
  - Mark the boundaries of each diphones
    - Cut each diphone out and create a diphone database

- **Synthesizing an utterance,**
  - Grab relevant sequence of diphones from database
  - Concatenate the diphones, doing slight signal processing at boundaries
  - Use signal processing to change the prosody (F0, energy, duration) of diphone sequence

Diphones

- **Mid-phone is more stable than edge**
- **Need ~|phones|^2 number of units**
  - Some combinations don't exist (hopefully)
  - ATT (Olive et al. 1998) system had 43 phones
    - 1849 possible diphones
    - Phonotactics ([h] only occurs before vowels), don't need to keep diphones across silence
    - Only 1172 actual diphones
  - May include stress, consonant clusters
    - So could have more
  - Lots of phonetic knowledge in design
- **Database relatively small (by today's standards)**
  - Around 8 megabytes for English (16 KHz 16 bit)

Summary: Diphone Synthesis

- **Well-understood, mature technology**
- **Augmentations**
  - Stress
  - Onset/coda
  - Demi-syllables
- **Problems:**
  - Signal processing still necessary for modifying durations
  - Source data is still not natural
  - Units are just not large enough; can't handle word-specific effects, etc

Problems with Diphone Synthesis

- **Signal processing methods like TD-PSOLA leave artifacts, making the speech sound unnatural**
- **Diphone synthesis only captures local effects**
  - But there are many more global effects (syllable structure, stress pattern, word-level effects)

Unit Selection Synthesis

Generalization of the diphone intuition

- **Larger units**
  - From diphones to sentences
- **Many many copies of each unit**
  - 10 hours of speech instead of 1500 diphones (a few minutes of speech)
- **Little or no signal processing applied to each unit**
  - Unlike diphones

Why Unit Selection Synthesis

Natural data solves problems with diphones

- **Diphone databases are carefully designed but:**
  - Speaker makes errors
  - Speaker doesn't speak intended dialect
  - Require database design to be right
- **If it's automatic**
  - Labeled with what the speaker actually said
  - Coarticulation, schwas, flaps are natural
- **"There's no data like more data"**
  - Lots of copies of each unit mean you can choose just the right one for the context
  - Larger units mean you can capture wider effects

Targets and Target Costs

- **A measure of how well a particular unit in the database matches the internal representation produced by the prior stages**
- **Features, costs, and weights**
- **Examples:**
  - /ih-t/ from stressed syllable, phrase internal, high F0, content word
  - /n-t/ from unstressed syllable, phrase final, low F0, content word
  - /dh-ax/ from unstressed syllable, phrase initial, high F0, from function word “the”

Slide: Paul Taylor

Join Costs

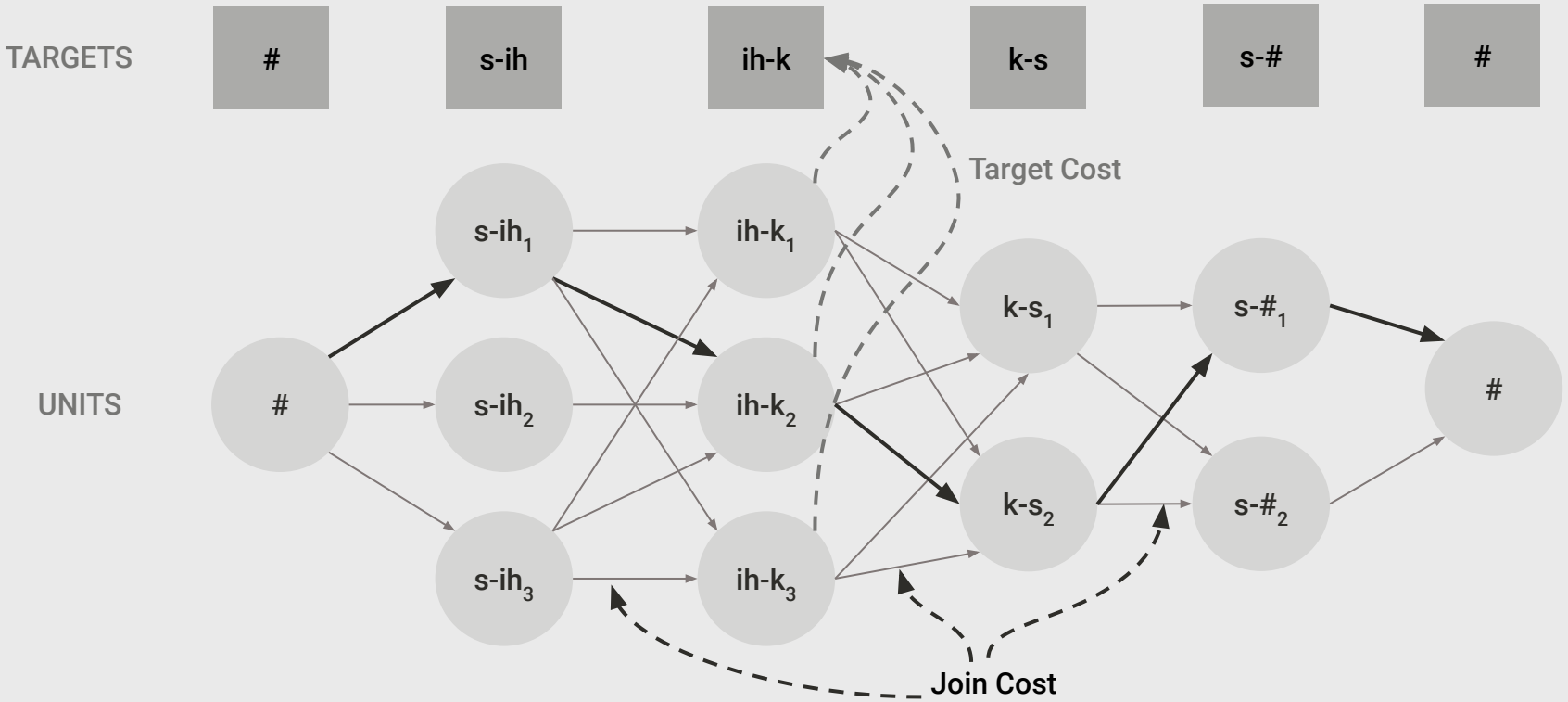
- **The join cost can be used for more than just part of search**
- **Can use the join cost for optimal coupling (Isard and Taylor 1991, Conkie 1996), i.e., finding the best place to join the two units.**
  - Vary edges within a small amount to find best place for join
  - This allows different joins with different units
  - Thus labeling of database (or diphones) need not be so accurate

Slide: Paul Taylor

Join Costs

- **The join cost can be used for more than just part of search**
- **Can use the join cost for optimal coupling , i.e., finding the best place to join the two units (Isard and Taylor 1991, Conkie 1996)**
  - Vary edges within a small amount to find best place for join
  - This allows different joins with different units
  - Thus labeling of database (or diphones) need not be so accurate

Unit Selection Search



Recap: Joining Units (+F0 + Duration)

- **For unit selection, just like diphone, need to join the units**
  - Pitch-synchronously
- **For diphone synthesis, need to modify F0 and duration**
  - For unit selection, in principle also need to modify F0 and duration of selection units
  - But in practice, if unit-selection database is big enough (commercial systems)
    - no prosodic modifications (selected targets may already be close to desired prosody)

Slide: Alan Black

Unit Selection Summary

Advantages:

- Quality is far superior to diphones
- Natural prosody selection sounds better

Disadvantages:

- Quality can be very bad in places
  - HCI problem: mix of very good and very bad is quite annoying
- Synthesis is computationally expensive
  - Can't synthesize everything you want:
  - Unit seleciton (unlike diphone synth) can't move emphasis
  - Unit selection gives good (but possibly incorrect) result

Slide: Richard Sproat