Speech Synthesis

Lecture 19

CS 753

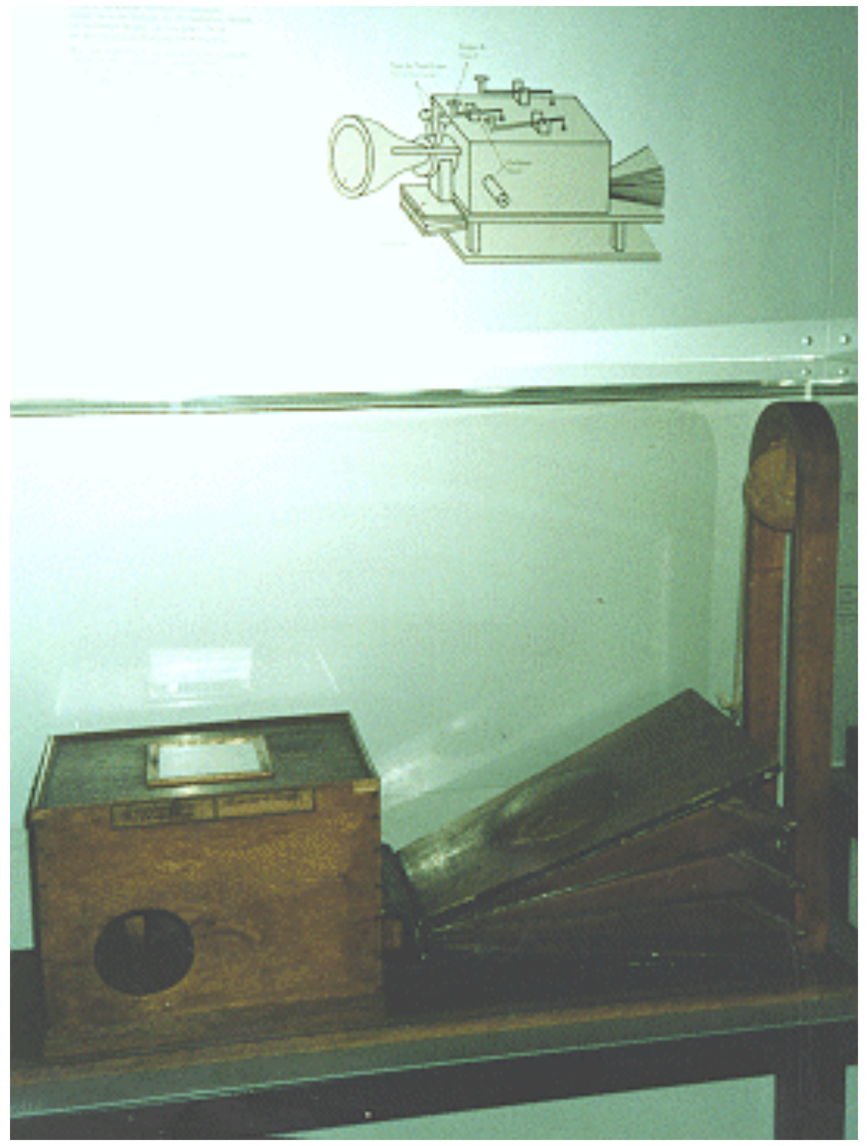
Instructor: Preethi Jyothi

Project Preliminary Report

- Preliminary project report will contribute towards 5% of your final grade. Deadline is on 27th October, 2019.
   - Define the following for your project: 1) Input-output behaviour of your system 2) Evaluation metric 3) At least two existing (or related) approaches to your problem **5 points**
   - Propose a model and an algorithm for the problem you're tackling and give detailed descriptions for both. Do not provide generic descriptions of the model. Describe precisely how it applies to your problem. **5 points**
   - Describe how much of your algorithm has been implemented. If you are using existing APIs/libraries, clearly demarcate which parts you will be implementing and for which parts you will rely on existing implementations. **5 points**
   - Describe the experiments you are planning to run. If you have already run any preliminary experiments, please describe them along with reporting your initial results. **5 points**

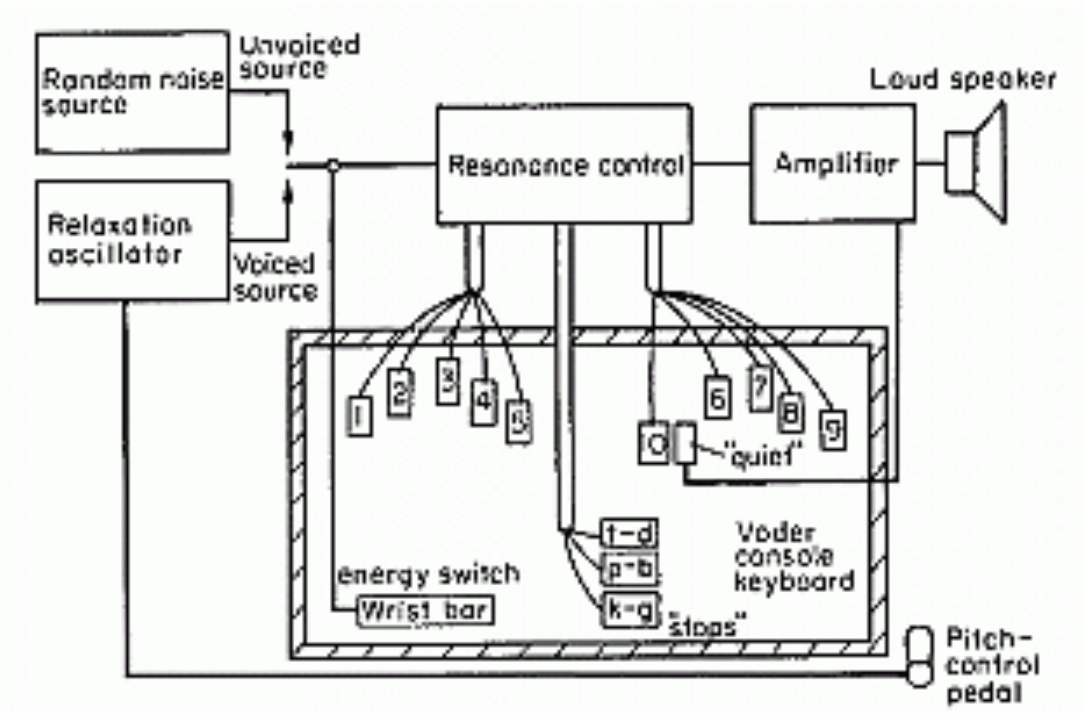
Text-To-Speech (TTS) Systems

Storied History

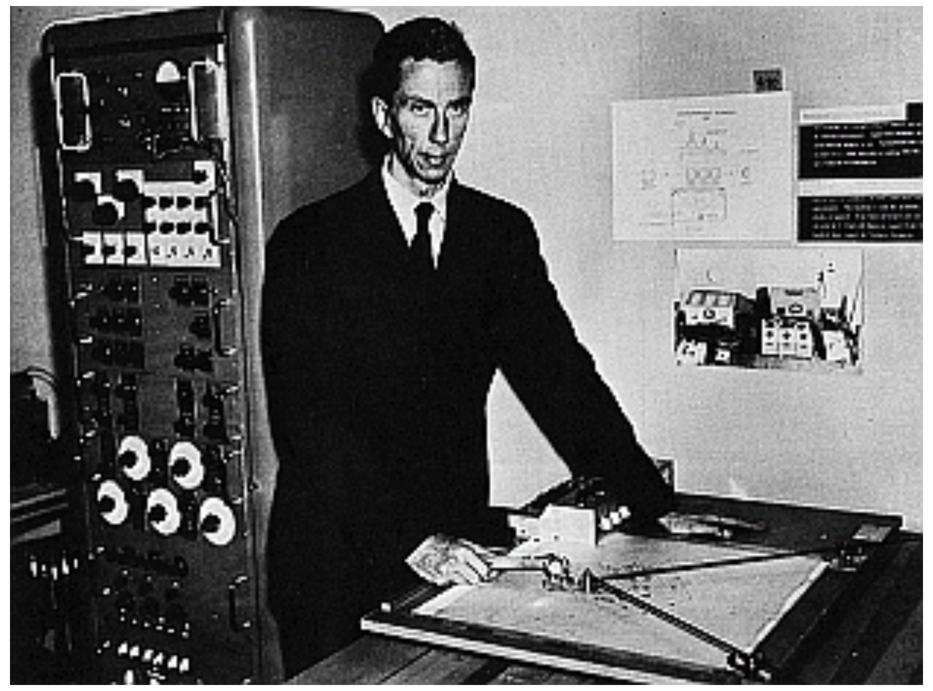


- Von Kempelen's speaking machine (1791)
  - Bellows simulated the lungs
  - Rubber mouth and nose; nostrils had to be covered with two fingers for non-nasals

- Homer Dudley's VODER (1939)
  - First device to synthesize speech sounds via electrical means



- Gunnar Fant's OVE formant synthesizer (1960s)
  - Formant synthesizer for vowels



- Computer-aided speech synthesis (1970s)
  - Concatenative (unit selection)
  - Parametric (HMM-based and NN-based)

All images from http://www2.ling.su.se/staff/hartmut/kemplne.htm

Speech synthesis or TTS systems

- Goal of a TTS system: Produce a natural-sounding high-quality speech waveform for a given word sequence
- TTS systems are typically divided into two parts:

  A. Linguistic specification

  B. Waveform generation

Current TTS systems

- Constructed using a large amount of speech data
- Referred to as corpus-based TTS systems
- Two prominent instances of corpus-based TTS:
    1. Unit selection and concatenation
    2. Statistical parametric speech synthesis

Unit Selection Synthesis

Unit selection synthesis or Concatenative speech synthesis

- Synthesize new sentences by selecting sub-word units from a database of speech
  - Optimal size of units? Diphones? Half-phones?

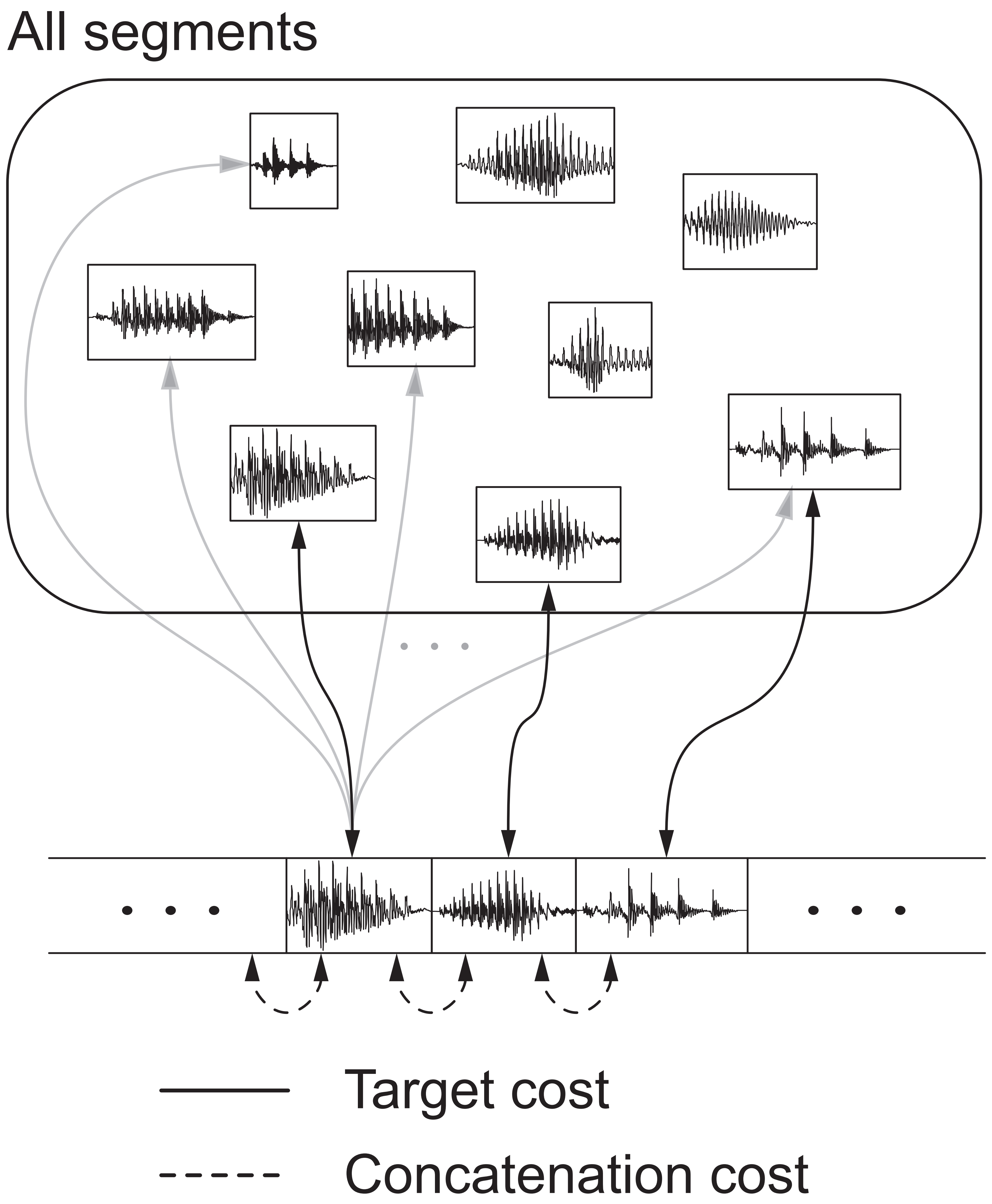


Image from Zen et al., "Statistical Parametric Speech Synthesis", SPECOM 2001

Unit selection synthesis

- Target cost between a candidate, u_i, and a target unit t_i:

$$C^{(t)}(t_i, u_i) = \sum_{j=1}^{p} w_j^{(t)} C_j^{(t)}(t_i, u_i),$$

- Concatenation cost between candidate units:

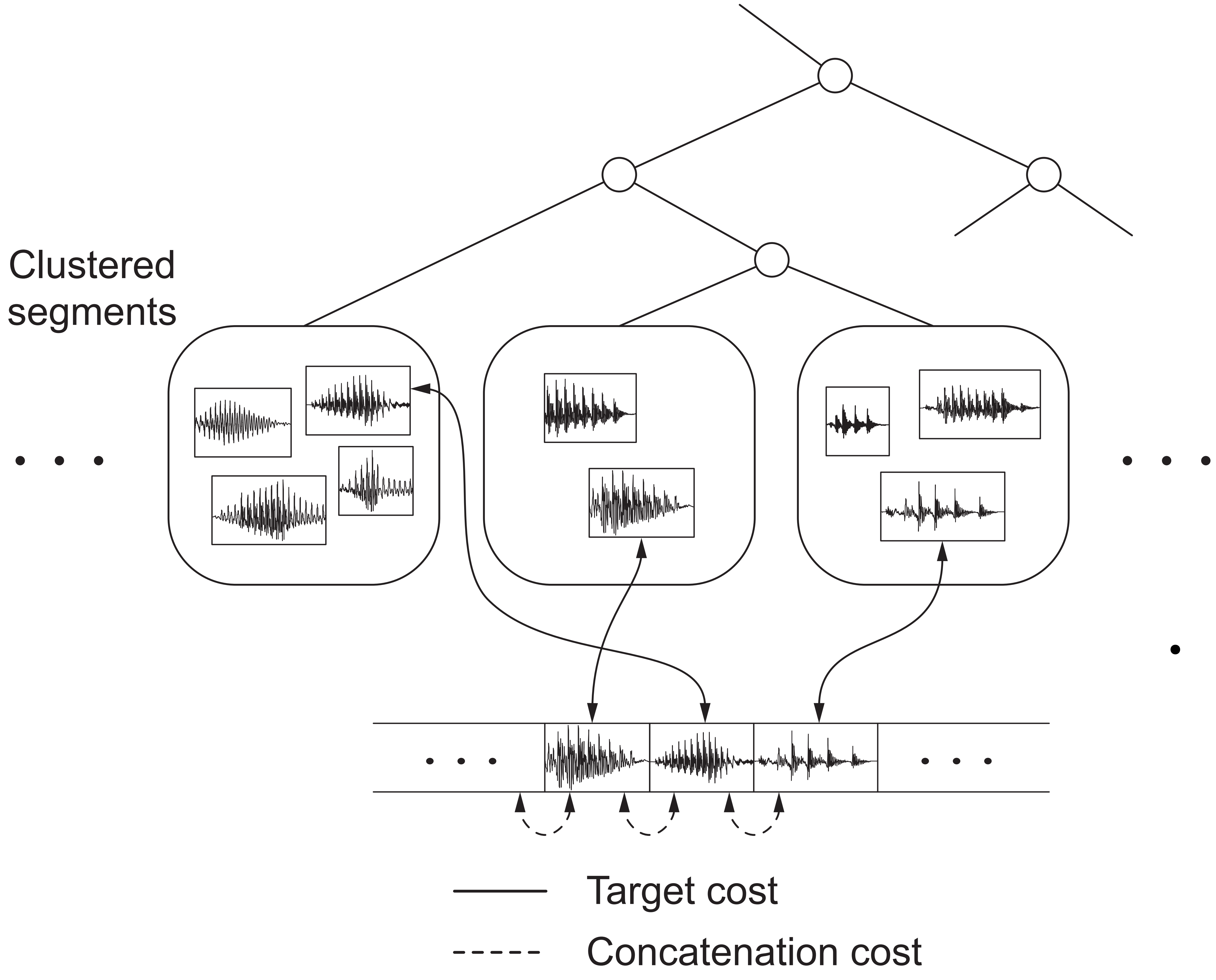
$$C^{(c)}(u_{i-1}, u_i) = \sum_{k=1}^{q} w_k^{(c)} C_k^{(c)}(u_{i-1}, u_i),$$

- Find string of units that minimises the overall cost:

$$\hat{u}_{1:n} = \arg\min_{u_{1:n}} \{C(t_{1:n}, u_{1:n})\}$$

$$C(t_{1:n}, u_{1:n}) = \sum_{i=1}^{n} C^{(t)}(t_i, u_i) + \sum_{i=2}^{n} C^{(c)}(u_{i-1}, u_i)$$

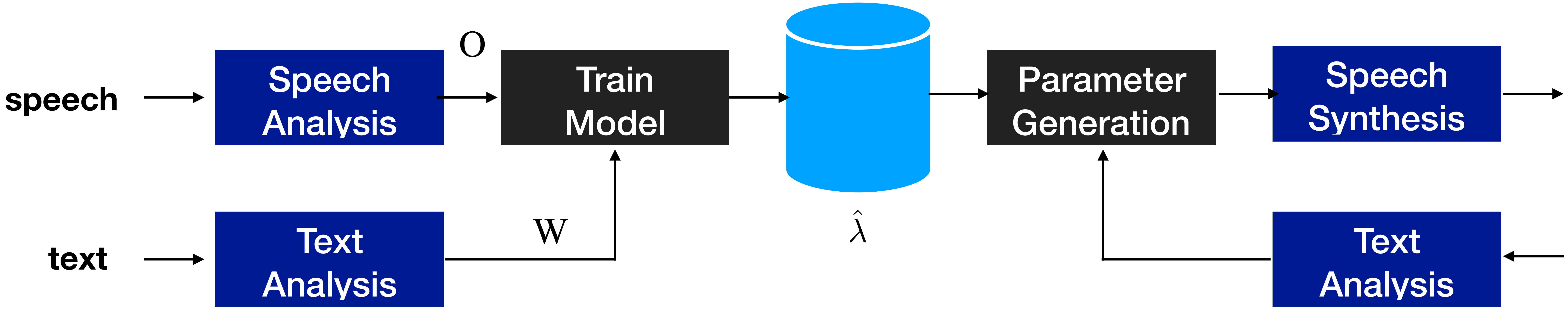
Unit selection synthesis



- Target cost is pre-calculated using a clustering method

Statistical Parametric Speech Synthesis

Parametric Speech Synthesis Framework

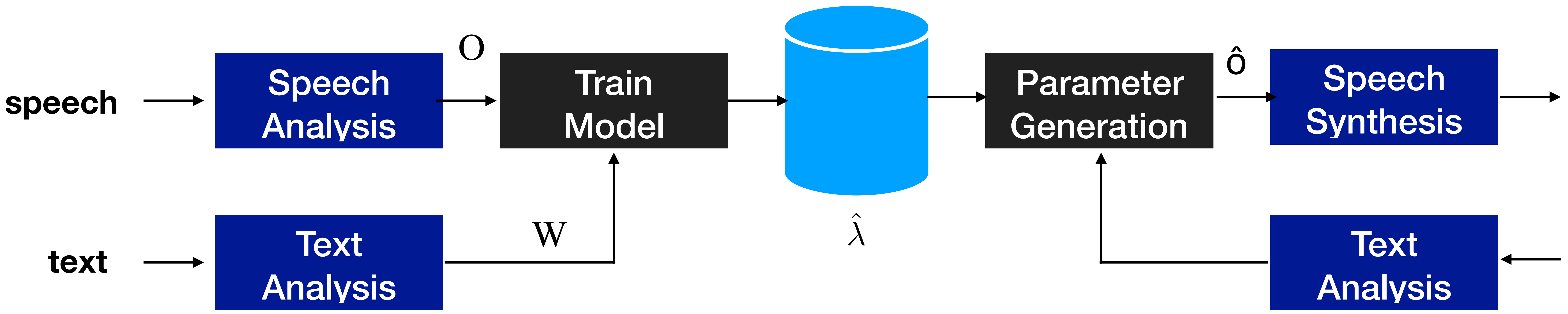


- Training
  - Estimate acoustic model given speech utterances (O), word sequences (W)*

$$\hat{\lambda} = \arg\max_{\lambda} p(O|W, \lambda)$$

* Here W could refer to any textual features relevant to the input text

Parametric Speech Synthesis Framework



- Training
  - Estimate acoustic model given speech utterances (O), word sequences (W)

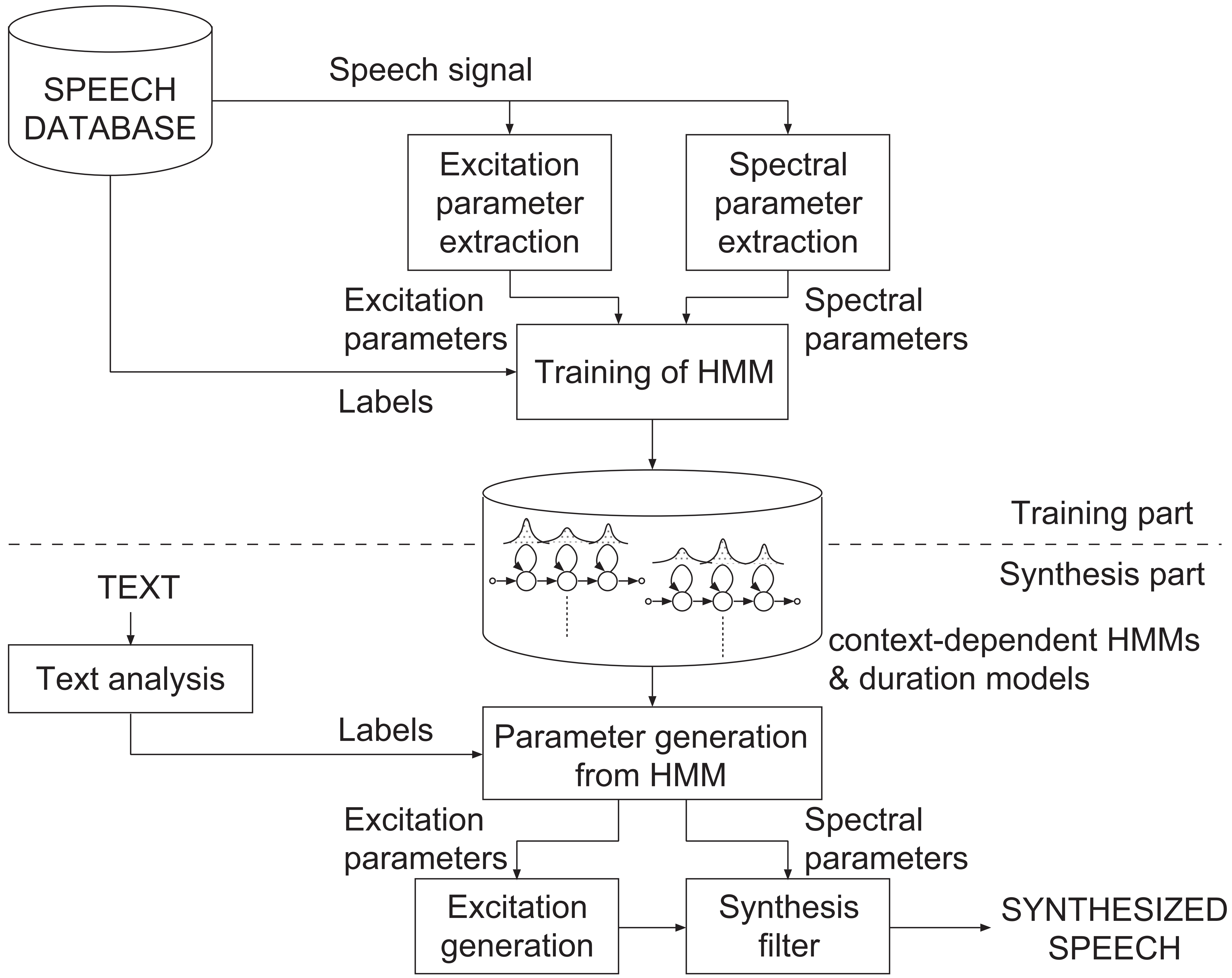
$$\hat{\lambda} = \arg\max_{\lambda} p(O|W, \lambda)$$

- Synthesis
  - Find the most probable ô from $\hat{\lambda}$ and a given word sequence to be synthesised, w

$$\hat{o} = \arg\max_{o} p(o|w, \hat{\lambda})$$

  - Synthesize speech from ô

HMM-based speech synthesis



Speech parameter generation

Generate the most probable observation vectors given the HMM and w:

$$\begin{aligned}
\hat{o} &= \arg\max_{o} p(o|w, \hat{\lambda}) \\
&= \arg\max_{o} \sum_{\forall q} p(o, q|w, \hat{\lambda}) \\
&\approx \arg\max_{o} \max_{q} p(o, q|w, \hat{\lambda}) \\
&= \arg\max_{o} \max_{q} p(o|q, \hat{\lambda}) p(q|w, \hat{\lambda})
\end{aligned}$$

Determine the best state sequence and outputs sequentially:

$$\hat{q} = \arg\max_{q} p(q|w, \hat{\lambda})$$

Let's explore this first

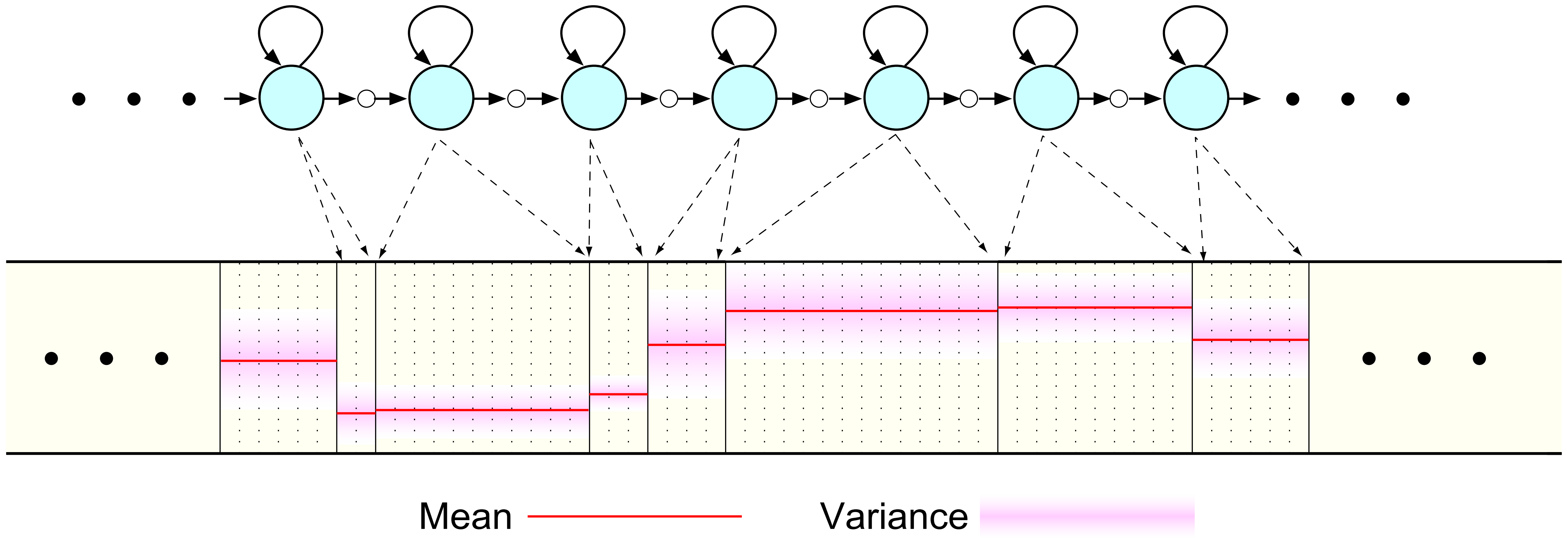
$$\boxed{\hat{o} = \arg\max_{o} p(o|\hat{q}, \hat{\lambda})}$$

Determining state outputs

$$\hat{o} = \arg\max_{o} p(o|\hat{q}, \hat{\lambda})$$
$$= \arg\max_{o} \mathcal{N}(o; \mu_{\hat{q}}, \Sigma_{\hat{q}})$$

where $\boldsymbol{o} = \left[\boldsymbol{o}_1^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$ is a state-output vector sequence to be generated, $\boldsymbol{q} = \{q_1, \ldots, q_T\}$ is a state sequence, and $\boldsymbol{\mu_q} = \left[\boldsymbol{\mu}_{q_1}^\top, \ldots, \boldsymbol{\mu}_{q_T}^\top\right]^\top$ is the mean vector for $\boldsymbol{q}$.

What would $\hat{o}$ look like?



Adding dynamic features to state outputs

$$\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top\right]^\top \quad \textbf{where} \quad \Delta\boldsymbol{c}_t = \boldsymbol{c}_t - \boldsymbol{c}_{t-1}$$

State output vectors contain both static (c_t) and dynamic (Δc_t) features

$$\begin{array}{ccccc}
\boldsymbol{o} & & \boldsymbol{W} & & \boldsymbol{c} \\
\left[\begin{array}{c}
\vdots \\ \hline
\boldsymbol{c}_{t-1} \\ \Delta\boldsymbol{c}_{t-1} \\ \hline
\boldsymbol{c}_t \\ \Delta\boldsymbol{c}_t \\ \hline
\boldsymbol{c}_{t+1} \\ \Delta\boldsymbol{c}_{t+1} \\ \hline
\vdots
\end{array}\right]
& = &
\left[\begin{array}{cccccc}
\cdots & \vdots & \vdots & \vdots & \vdots & \cdots \\ \hline
\cdots & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \cdots \\
\cdots & -\boldsymbol{I} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \cdots \\ \hline
\cdots & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \cdots \\
\cdots & \mathbf{0} & -\boldsymbol{I} & \boldsymbol{I} & \mathbf{0} & \cdots \\ \hline
\cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \cdots \\
\cdots & \mathbf{0} & \mathbf{0} & -\boldsymbol{I} & \boldsymbol{I} & \cdots \\ \hline
\cdots & \vdots & \vdots & \vdots & \vdots & \cdots
\end{array}\right]
& &
\left[\begin{array}{c}
\vdots \\ \hline
\boldsymbol{c}_{t-2} \\ \hline
\boldsymbol{c}_{t-1} \\ \hline
\boldsymbol{c}_t \\ \hline
\boldsymbol{c}_{t+1} \\ \hline
\vdots
\end{array}\right]
\end{array}$$

o and c arranged in matrix form

Speech parameter generation

- Introducing dynamic feature constraints:

$$\hat{o} = \arg\max_{o} p(o|\hat{q}, \hat{\lambda}) \text{ where } o = Wc$$

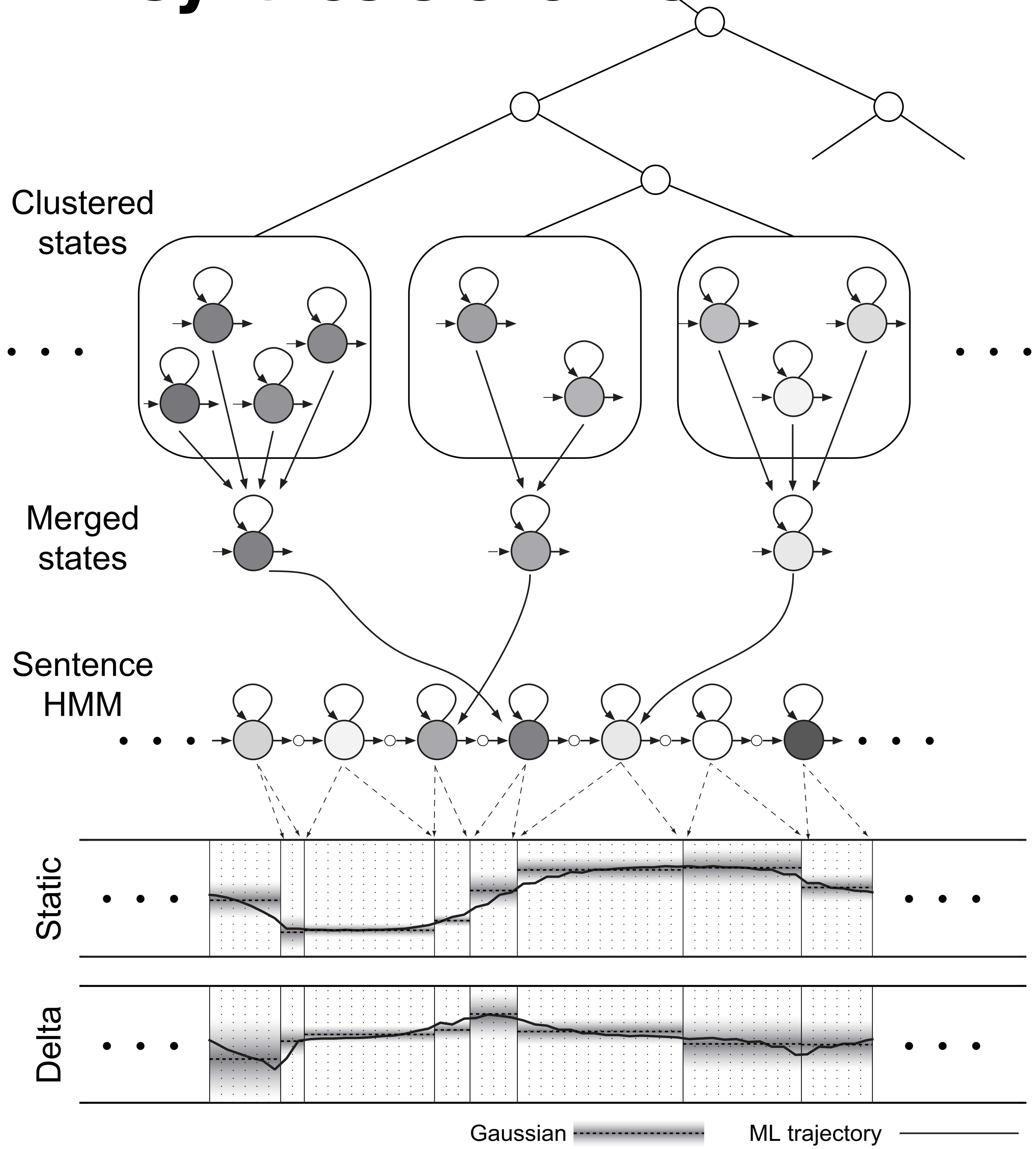
- If the output distributions are single Gaussians:

$$p(o|\hat{q}, \hat{\lambda}) = \mathcal{N}(o; \mu_{\hat{q}}, \Sigma_{\hat{q}})$$

- Then, by setting $\partial \log \mathcal{N}(o; \mu_{\hat{q}}, \Sigma_{\hat{q}}) / \partial c = 0$ we get:

$$W^T \Sigma_{\hat{q}}^{-1} W c = W^T \Sigma_{\hat{q}}^{-1} \mu_{\hat{q}}$$

Synthesis overview



Speech parameter generation

Generate the most probable observation vectors given the HMM and w:

$$\begin{aligned}
\hat{o} &= \arg\max_{o} p(o|w,\hat{\lambda}) \\
&= \arg\max_{o} \sum_{\forall q} p(o,q|w,\hat{\lambda}) \\
&\approx \arg\max_{o} \max_{q} p(o,q|w,\hat{\lambda}) \\
&= \arg\max_{o} \max_{q} p(o|q,\hat{\lambda})P(q|w,\hat{\lambda})
\end{aligned}$$

Determine the best state sequence and outputs sequentially:

Let's explore this next

$$\hat{q} = \arg\max_{q} p(q|w,\hat{\lambda})$$

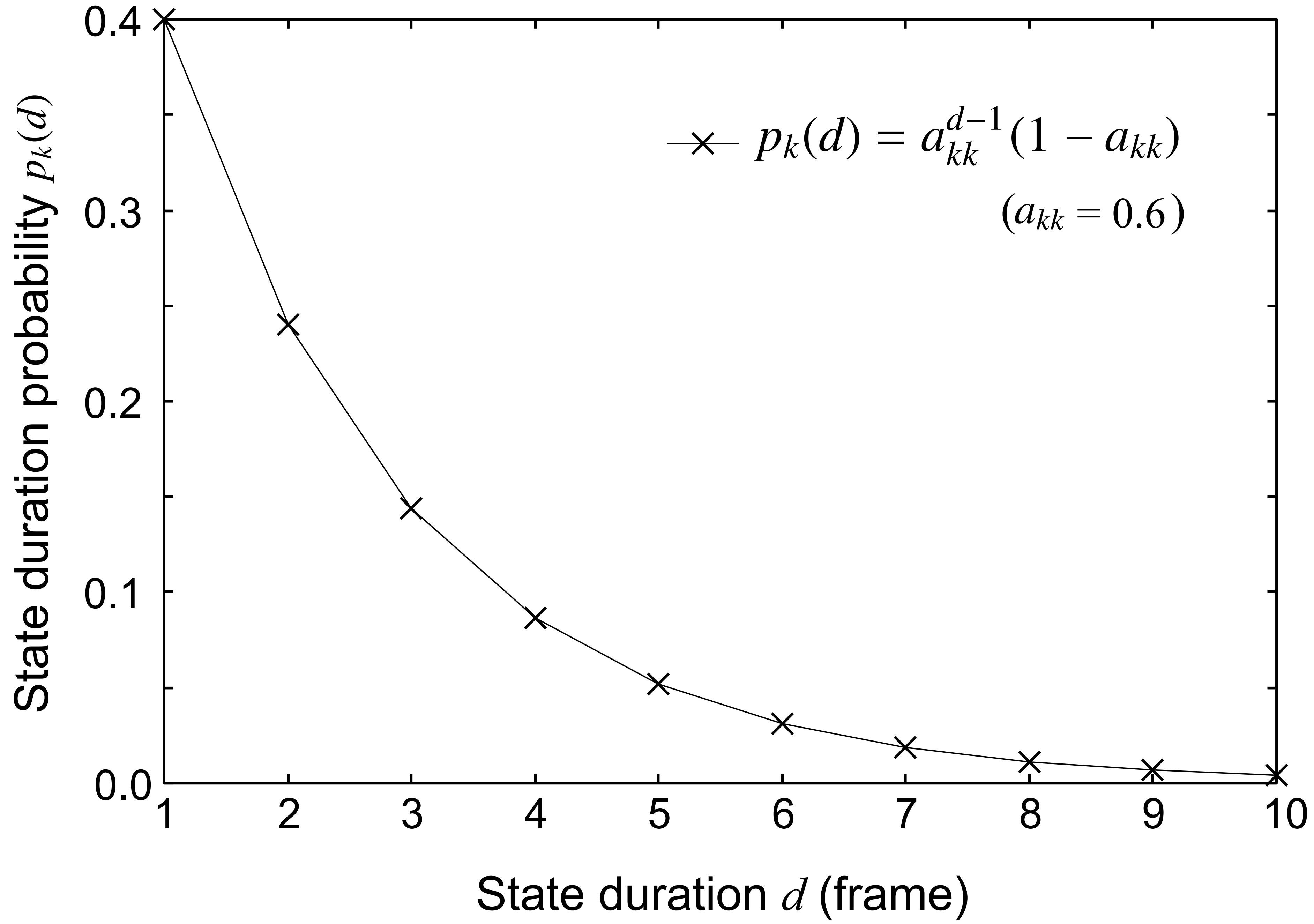
$$\hat{o} = \arg\max_{o} p(o|\hat{q},\hat{\lambda})$$

Duration modeling

- How are durations modelled within an HMM?
  - Implicitly modelled by state self-transition probabilities

$$p_k(d) = a_{kk}^{d-1} \cdot (1 - a_{kk})$$

  - PMFs of state durations are geometric distributions



- State durations are determined by maximising:

$$\log P(\boldsymbol{d} \mid \lambda) = \sum_{j=1}^{N} \log p_j(d_j),$$

  - What would this solution look like if the PMFs of state durations are geometric distributions?

Explicit modeling of state durations

- Each state duration is explicitly modelled as a single Gaussian. The mean $\xi(i)$ and variance $\sigma^2(i)$ of duration density of state i:

$$\xi(i) = \frac{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T} \chi_{t_0,t_1}(i)(t_1 - t_0 + 1)}{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T} \chi_{t_0,t_1}(i)},$$

$$\sigma^2(i) = \frac{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T} \chi_{t_0,t_1}(i)(t_1 - t_0 + 1)^2}{\sum_{t_0=1}^{T}\sum_{t_1=t_0}^{T} \chi_{t_0,t_1}(i)} - \xi^2(i)$$

where

$$\chi_{t_0,t_1}(i) = (1 - \gamma_{t_0-1}(i)) \cdot \prod_{t=t_0}^{t_1} \gamma_t(i) \cdot (1 - \gamma_{t_1+1}(i)),$$

and $\gamma_t(i)$ is the probability of being in state i at time t

Determining state durations

During synthesis, for a given speech length T, the goal is to maximize:

$$\log P(\mathbf{d}|\lambda, T) = \sum_{k=1}^{K} \log p_k(d_k) \qquad \text{… (1)}$$

under the constraint that $T = \sum_{k=1}^{K} d_k$

We saw that each duration density $p_k(d_k)$ can be modelled as a single Gaussian $\mathcal{N}(\cdot; \xi_k, \sigma_k^2)$

State durations, $d_k,\ k = 1 \ldots K$, which maximise (1) are given by:

$$\begin{aligned} d_k &= \xi(k) + \rho \cdot \sigma^2(k) \\ \rho &= \left(T - \sum_{k=1}^{K} \xi(k)\right) \Big/ \sum_{k=1}^{K} \sigma^2(k) \end{aligned}$$

Speaking rate

Synthesis using duration models

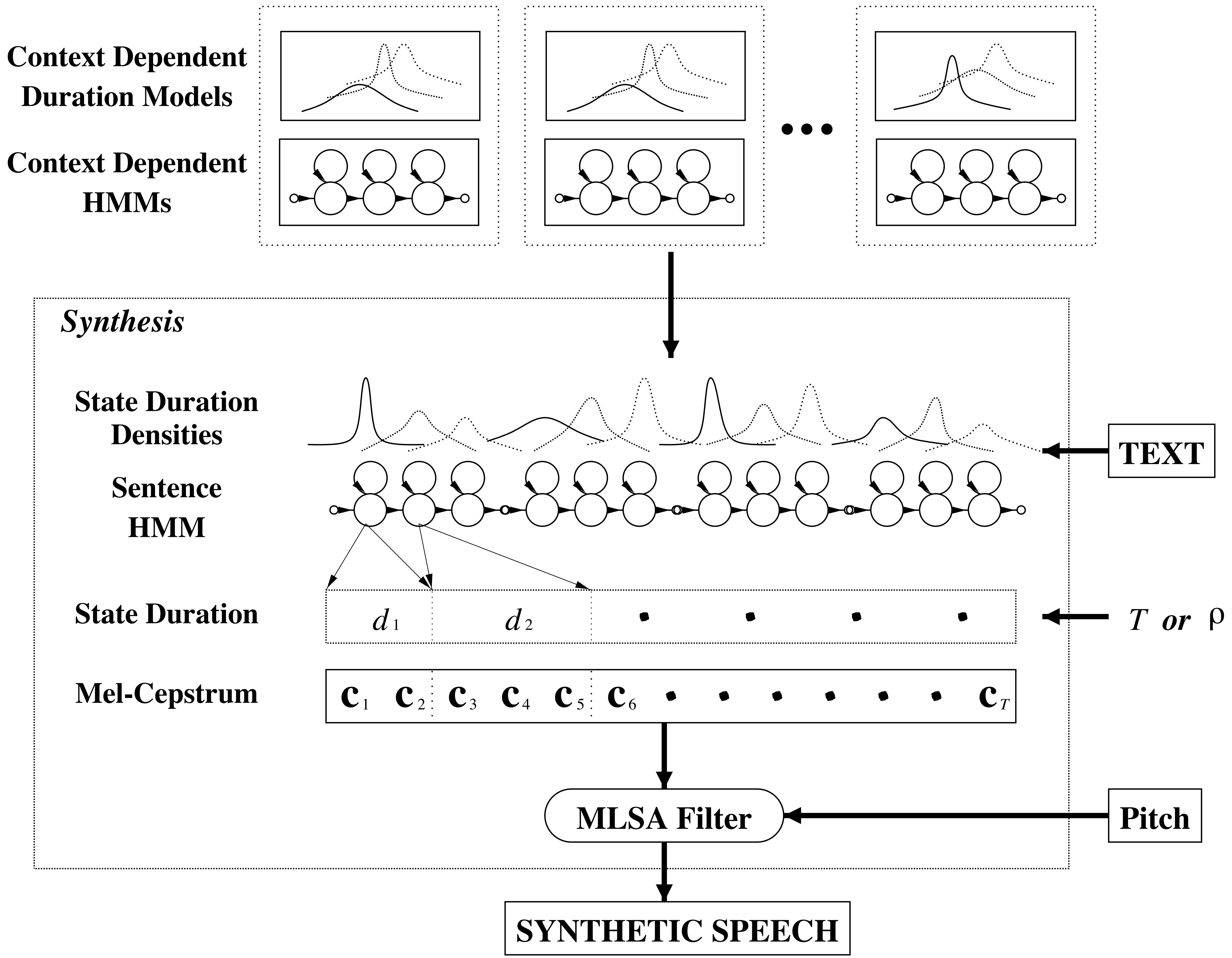
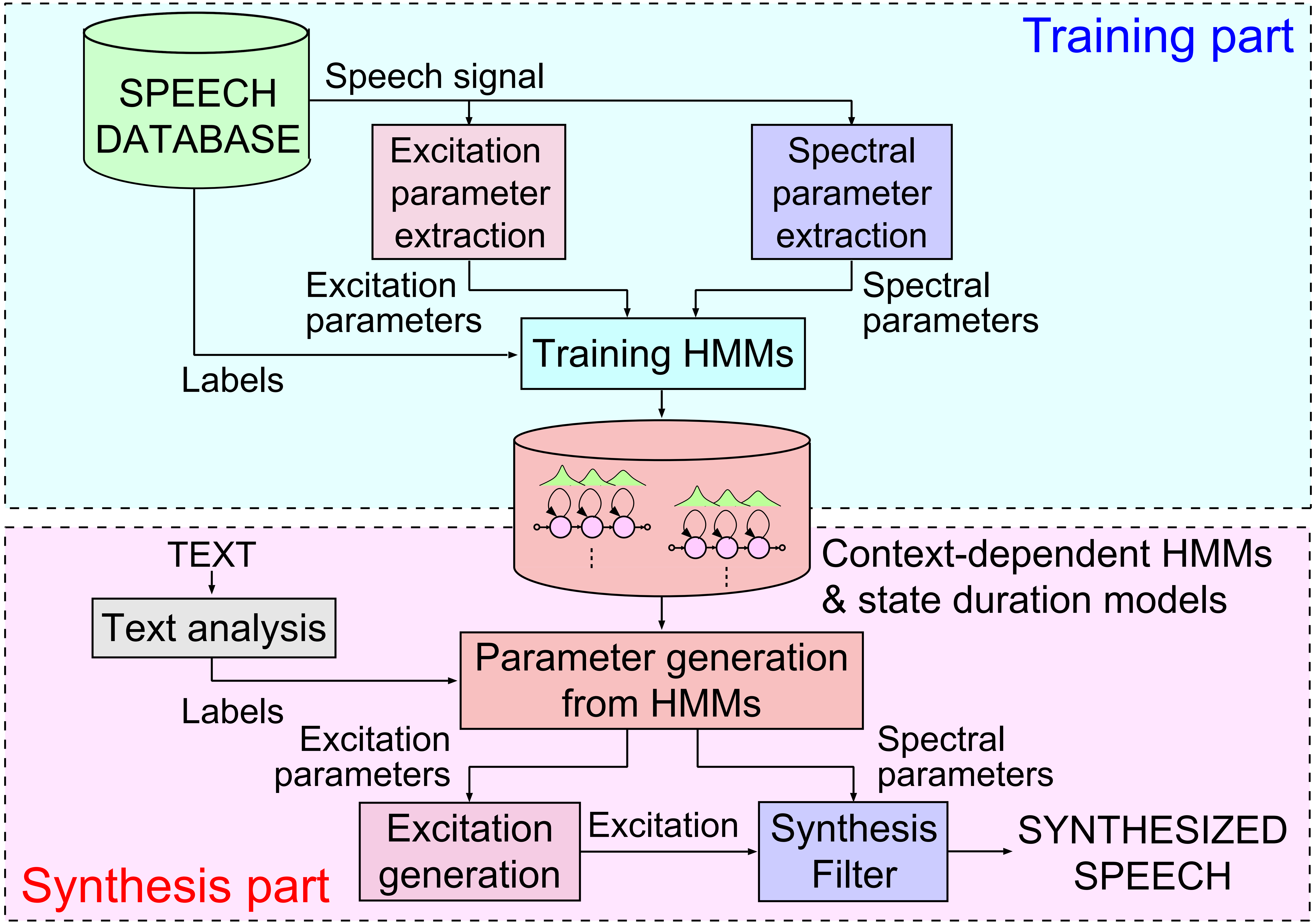


Image from Yoshimura et al., “Duration modelling for HMM-based speech synthesis”, ICSLP ‘98

Recap: HMM-based speech synthesis



DNN-based speech synthesis

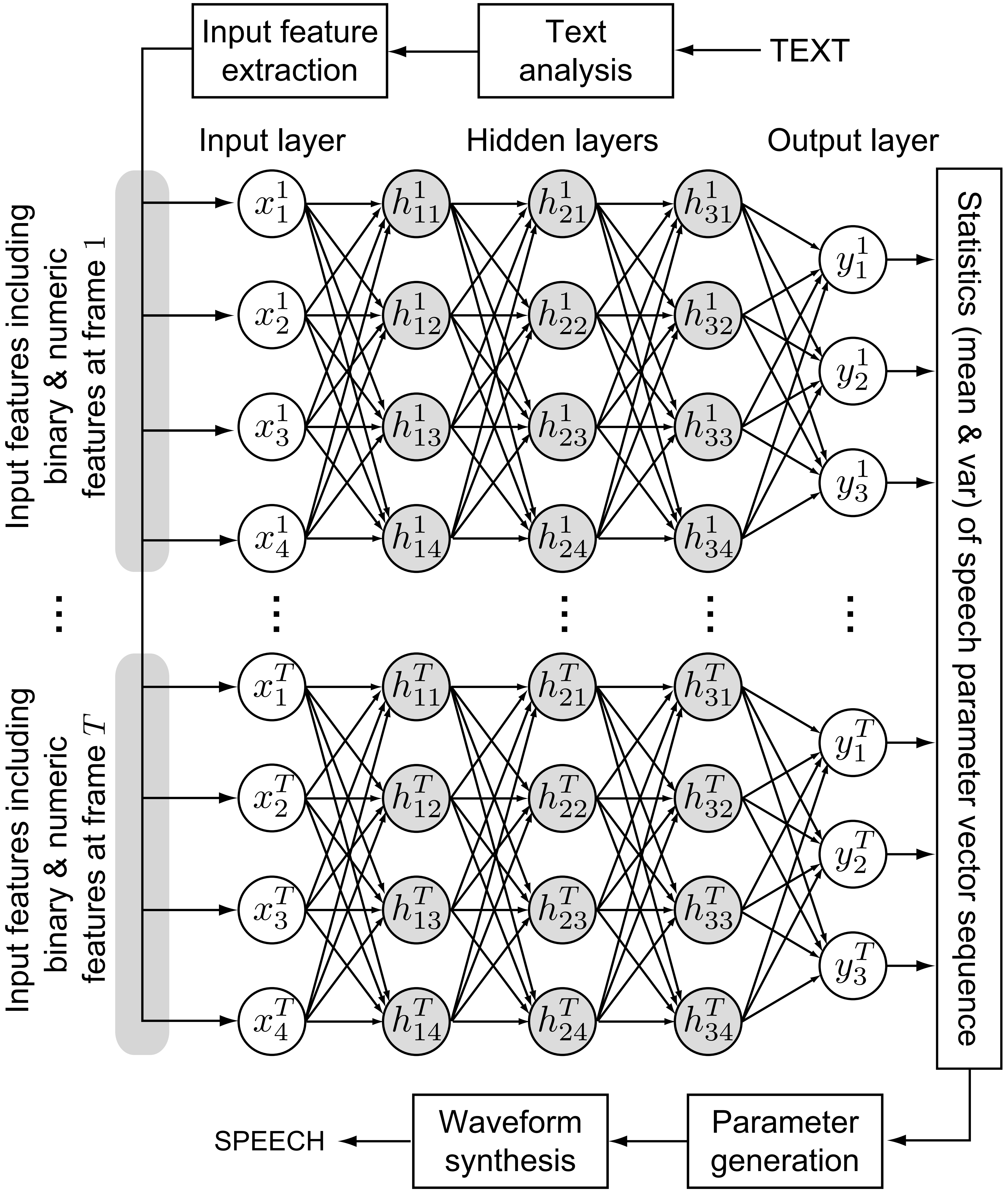


Image from Zen et al., “Statistical Parametric Speech Synthesis using DNNs”, 2014

DNN-based speech synthesis

- Input features about linguistic contexts, numeric values (# of words, duration of the phoneme, etc.)
- Output features are spectral and excitation parameters and their delta values
- Listening test results

Table 1. Preference scores (%) between speech samples from the HMM and DNN-based systems. The systems which achieved significantly better preference at $p < 0.01$ level are in the bold font.

HMM (α)	DNN (#layers × #units)	Neutral	p value	z value
15.8 (16)	**38.5** (4 × 256)	45.7	$< 10^{-6}$	-9.9
16.1 (4)	**27.2** (4 × 512)	56.8	$< 10^{-6}$	-5.1
12.7 (1)	**36.6** (4 × 1 024)	50.7	$< 10^{-6}$	-11.5

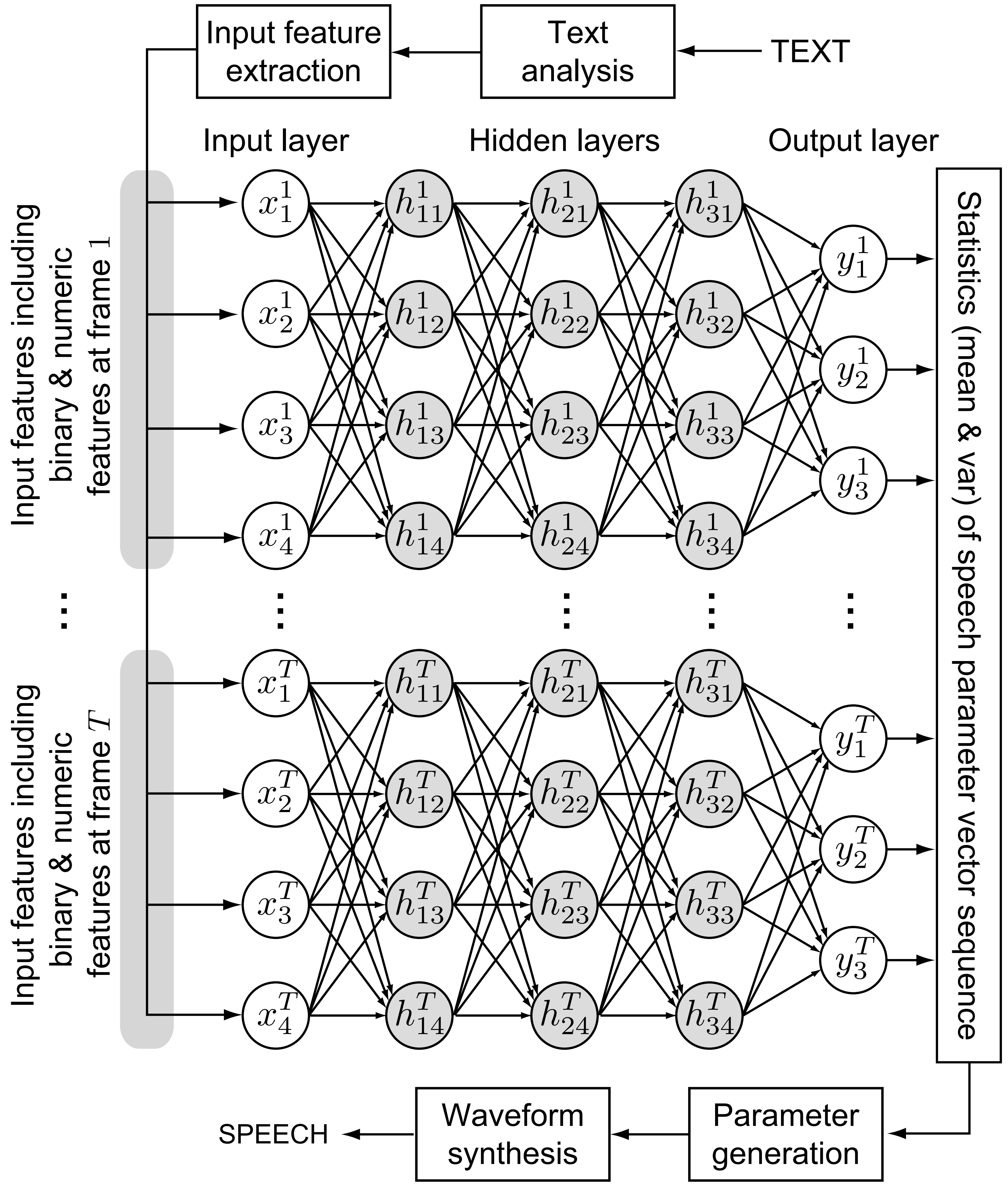


Image from Zen et al., "Statistical Parametric Speech Synthesis using DNNs", 2014

RNN-based speech synthesis

- Access long range context in both forward backward directions using biLSTMs
- Inference is expensive; inherently have large latency

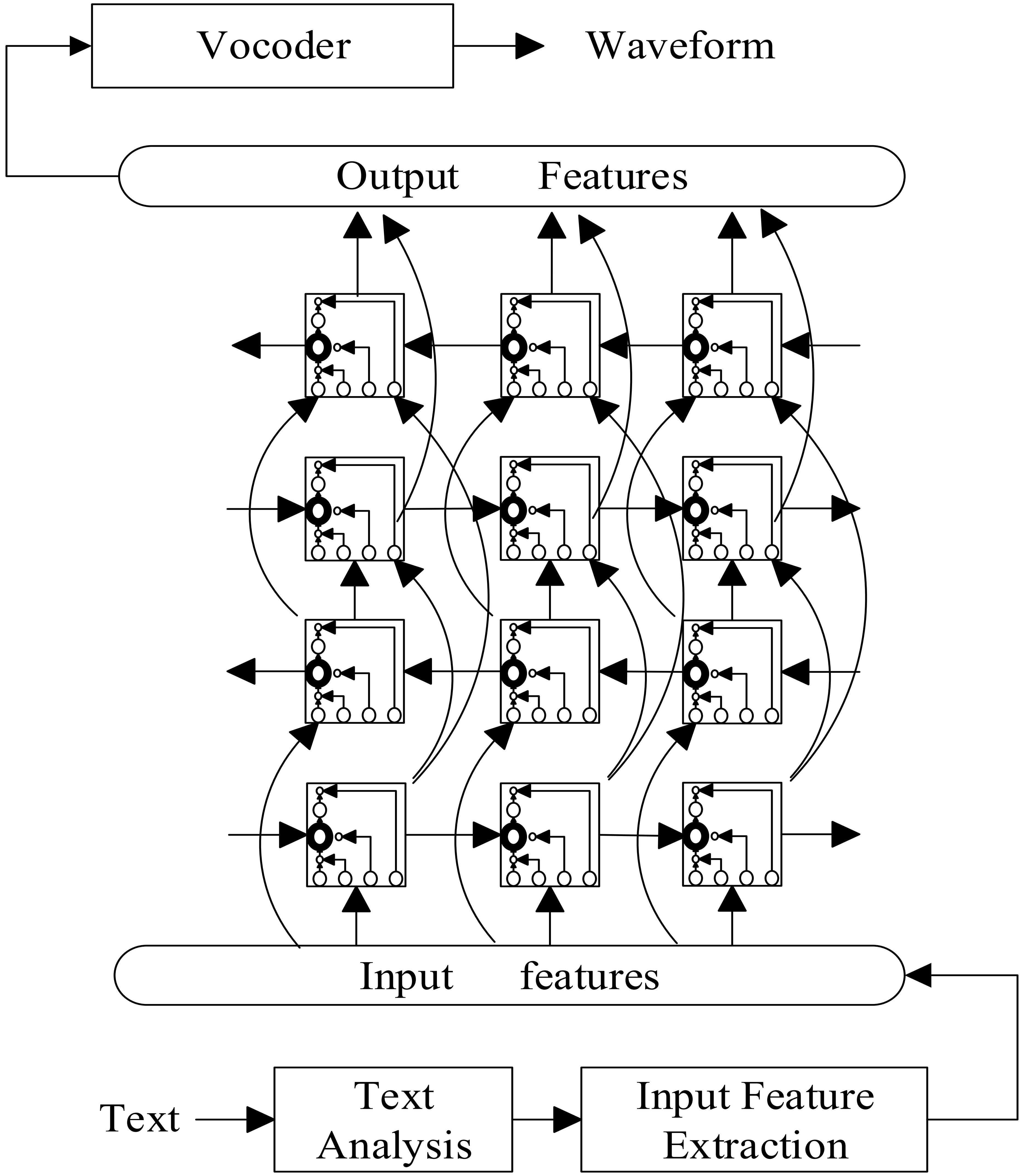


Image from Fan et al., "TTS synthesis with BLSTM-based RNNs", 2014