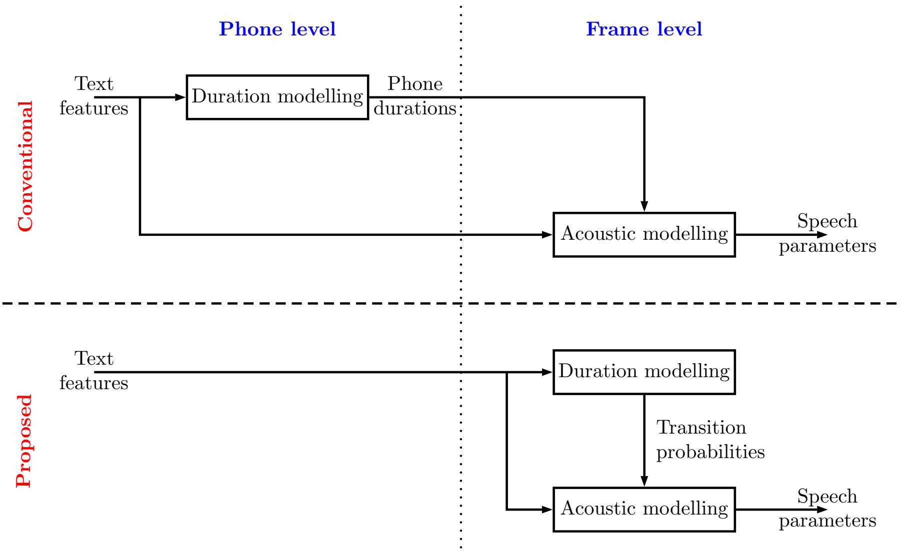
Non-parametric duration modelling for speech synthesis with a joint model of acoustics and duration

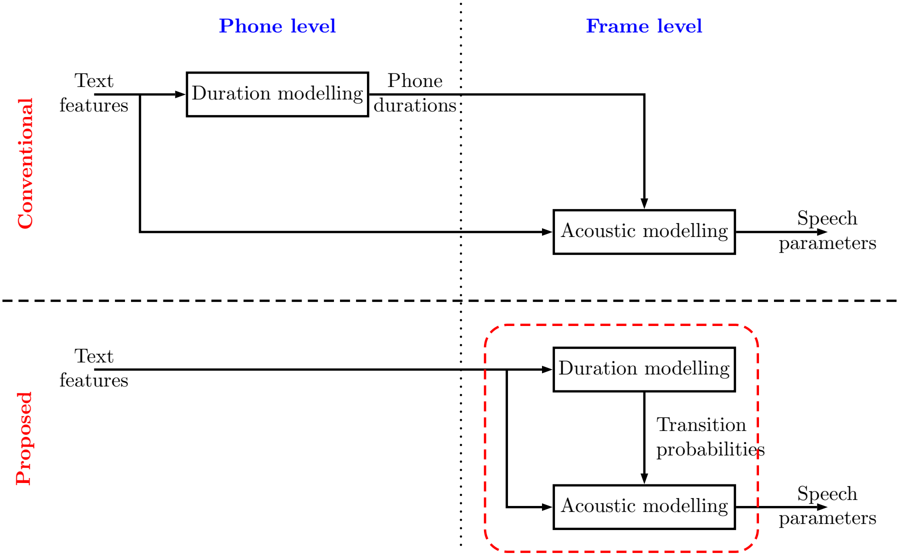
Gustav Eje Henter[1], Srikanth Ronanki[2], Oliver Watts[2], and Simon King[2]

[1]Digital Content and Media Sciences Research Division, National Institute of Informatics, Tokyo
[2]The Centre for Speech Technology Research (CSTR), The University of Edinburgh, UK

Graphical overview



Graphical overview



Key takeaways

- Innovations
  1. Train an RNN/DNN to predict per-frame transition probabilities
  2. Generate durations using median or other distribution quantiles
- Advantages
  - *Non-parametric – can model any duration distribution shape!*
  - Predicts acoustics and durations in tandem
  - Is a proper hidden semi-Markov model

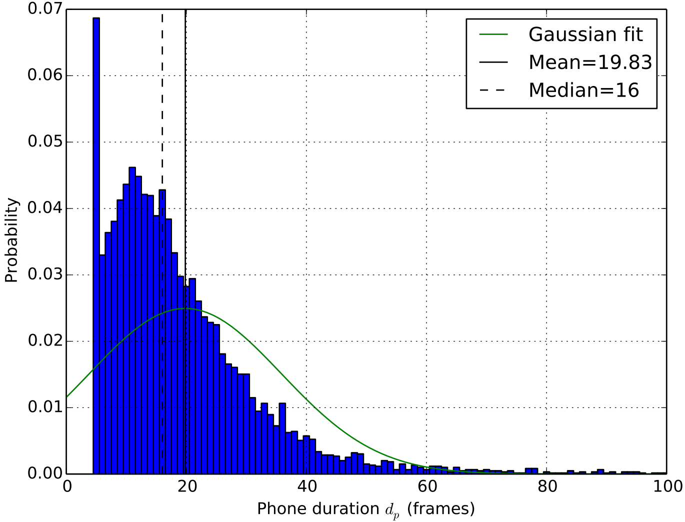
Outline

Motivation

- Prosody remains a major shortcoming of TTS
  - Duration is an important prosodic component
- State-of-the-art (Gaussian) duration models:
  - Allow non-positive durations
  - Do not sum to one on the integers (unnormalised)
  - Do not account for skewness
  - Are separate from the acoustic model

Real durations

Forced-aligned durations from dataset vs. fitted Gaussian



Statistical TTS

Statistical parametric speech synthesis requires three components:

1. A stochastic distribution family $f_D(d;\, \boldsymbol{\theta})$ for durations D
2. A machine-leaning predictor $\boldsymbol{\theta}(\boldsymbol{l})$
   - $\boldsymbol{l}$ are text-derived linguistic features
   - Predicts how duration distributions depend on text
   - Is learned from training data (statistical)
3. A duration-generation principle
   - Mean-based generation $\widehat{d} = \mathbb{E}(D \mid \boldsymbol{l})$

HMM-based TTS

Speech is generated by a hidden Markov model (HMM)

- Hidden-state models specified by:
  - Emissions: $f_{\boldsymbol{O} \mid S}(\boldsymbol{o} \mid s)$ (acoustic observations $\boldsymbol{o}$)
  - Transition probability: $\mathbb{P}(S_{t+1} = s + 1 \mid S_t = s)$ (durations)
    - State transitions follow a Markov process
  - State S_t tracks sub-phone time evolution
- Training (EM-algorithm) is linear in sequence length

HMM-based durations

1. Geometric duration distribution $f_D(d;\, a) = a\,(1-a)^{d-1}$
   - Implicit consequence of fixed HMM transition probability a
   - Memoryless (unrealistic)
2. Regression tree (RT) predictor $a\,(\boldsymbol{l})$
3. Mean-based generation $\widehat{d} = \mathbb{E}\,(D \mid \boldsymbol{l}) \propto \frac{1}{a(\boldsymbol{l})}$

HSMM-based TTS

Change to a hidden-semi Markov model (HSMM) (Zen et al., 2004)

- Model specified by:
  - Emissions: $f_{\boldsymbol{O} \mid S}(\boldsymbol{o} \mid s)$
    - Unchanged
  - Transition probability: $\mathbb{P}(S_{t+1} = s + 1 \mid S_t = s,\ n_t)$
    - Can now depend on n_t, time spent in current state
    - This is a semi-Markov process
- Training complexity is now quadratic in sequence length

HSMM-based durations

1. Any parametric distribution $f_D\left(d;\, \boldsymbol{\theta}\right)$ possible!
    - Gaussian distribution $f_D\left(d;\, \boldsymbol{\theta}\right) = f_{\mathcal{N}}\left(d;\, \mu,\, \sigma^2\right)$ standard in HTS (Zen et al., 2007)
    - Log-normal (Campbell, 1989) or gamma (Huber, 1990)
2. Regression tree (RT) predictor $\boldsymbol{\theta}\left(\boldsymbol{l}\right)$
    - Unchanged
3. Mean-based generation $\widehat{d} = \mathbb{E}\left(D \mid \boldsymbol{l}\right) = \widehat{\mu}\left(\boldsymbol{l}\right)$
    - Unchanged

NN-based durations

1. Gaussian distribution $f_D(d;\, \boldsymbol{\theta}) = f_{\mathcal{N}}(d;\, \mu,\, \sigma^2)$
   - Unchanged
2. Deep or recurrent neural network $\mu(\boldsymbol{l})$
   - DNNs/RNNs are more successful practical predictors
   - Typically, only μ is predicted (minimum MSE)
3. Mean-based generation $\widehat{d} = \mathbb{E}(D \mid \boldsymbol{l}) = \widehat{\mu}(\boldsymbol{l})$
   - Unchanged

Note: Data is forced-aligned using HMM/HSMM before training

Approaches in review

TTS type	$f_D(d; \boldsymbol{\theta})$	**Level**	**Pred.** $\boldsymbol{\theta}(\boldsymbol{l})$	**Generation**
Formant	-	Phone	-	Rule
Concat.	-	Phone	-	Exemplar
HMM	Geom.	State	RT	Mean
HSMM	Param.	State	RT	Mean
NN	Gauss.	State	NN	Mean

Approaches in review

TTS type	$f_D(d;\,\boldsymbol{\theta})$	**Level**	**Pred.** $\boldsymbol{\theta}(\boldsymbol{l})$	**Generation**
Formant	-	Phone	-	Rule
Concat.	-	Phone	-	Exemplar
HMM	Geom.	State	RT	Mean
HSMM	Param.	State	RT	Mean
NN	Gauss.	State	NN	Mean
Proposed	Non-par.	$\leq$Frame	NN	Quantile

Proposed approach

1. General categorical distribution $f_D(d)$
   - Not restricted to a specific parametric form
2. Deep or recurrent neural network
   - Predicts a transition probability for each time unit (e.g., frame)
   - Runs in tandem with acoustic model
3. Quantile-based generation
   - Can be computed using $\mathbb{P}(D \leq d)$, the left tail of f_D, only
   - Median duration: Special case more probable than mean
   - Benefits from statistical robustness (Henter et al., 2016)

Outline

Preliminaries

- $p \in \{1, \ldots, P\}$ is a phone/state index
- $t \in \{1, \ldots, T\}$ is a time-step (frame) index
- D_p is the (stochastic) duration of phone/state p
    - Outcome values $d_p \in \mathbb{Z} > 0$
- $\boldsymbol{l}_p$ collects the per-phone linguistic features
- The task is to generate durations: $(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_P) \rightarrow \left(\widehat{d}_1, \ldots, \widehat{d}_P\right)$

Conventional setup

- Phone-level dataset $\mathcal{D}_p = ((\boldsymbol{l}_1, \ldots, \boldsymbol{l}_P), (d_1, \ldots, d_P))$
  - $\boldsymbol{L}_p$ denotes the linguistic information influencing predictor at p
  - $\boldsymbol{L}_p = (\boldsymbol{l}_1, \ldots, \boldsymbol{l}_p)$ for a unidirectional RNN
- Phone-level DNN/RNN $d(\boldsymbol{L}_p;\ \boldsymbol{W})$ predicts duration directly
  - NN weights $\boldsymbol{W}$ chosen to minimise MSE prediction error

$$\widehat{\boldsymbol{W}}(\mathcal{D}_p) = \operatorname*{argmin}_{\boldsymbol{W}} \sum_{p \in \mathcal{D}_p} (d_p - d(\boldsymbol{L}_p;\ \boldsymbol{W}))^2$$

  - The theoretical MSE minimiser is the expected duration
- Frame-level acoustic modelling is a separate stage

Frame-level data

- Frame-level sequence of linguistic features

$$\boldsymbol{L}_t = (\boldsymbol{l}_1, \ldots, \boldsymbol{l}_t) = \left(\boldsymbol{l}_{p(1)}, \ldots, \boldsymbol{l}_{p(t)}\right)$$

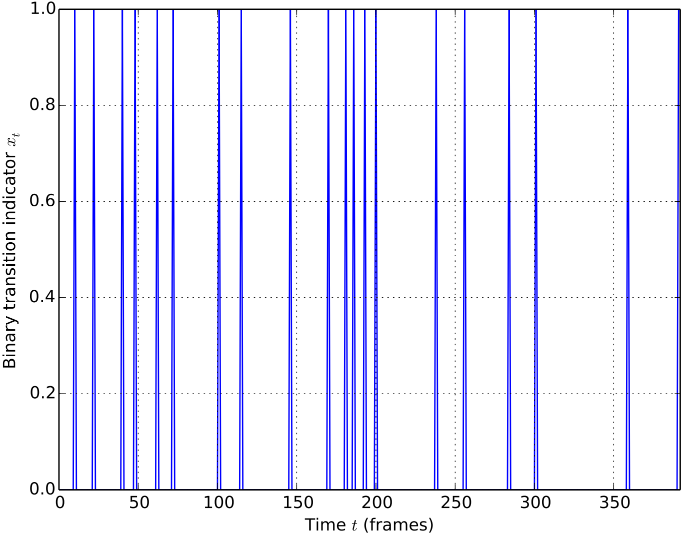
  - $p(t)$ is the current phone at frame t
  - t_0 is the end frame of the previous phone
  - The current phone has lasted $n_t = t - t_0$ frames
- Define per-frame indicator variables

$$x_t = \mathbb{I}\left(n_t = d_{p(t)}\right)$$

  - Equal one if t is the last frame of phone $p(t)$, and zero otherwise
- Frame-level dataset $\mathcal{D}_t = (\boldsymbol{L}_T, (x_1, \ldots, x_T))$

Example

Example of binary x_t sequence from database utterance



Transition probabilities

- Idea: Consider the transition probability

$$\pi_t = \pi\left(\boldsymbol{L}_t\right) = \mathbb{P}\left(D_p = n_t \mid D_p \geq n_t,\, \boldsymbol{L}_t\right)$$

  - $1 - \pi_t$ is the probability to remain in the same phone/state
- This defines an unambiguous, proper duration distribution

$$\mathbb{P}\left(D_p = n_t \mid \boldsymbol{L}_t\right) = \pi\left(\boldsymbol{L}_t\right) \prod_{t'=t_0+1}^{t_0+n_t-1} \left(1 - \pi\left(\boldsymbol{L}_{t'}\right)\right)$$

  if and only if
  - $\pi_t \in [0,\, 1]\ \forall t$
  - $\prod_{t'=t_0+1}^{\infty} \left(1 - \pi_{t'}\right) = 0$ when $p\left(t'\right)$ constant
- All distributions on the positive integers writeable like this

Predicting transitions

- Frame-level RNN $x\left(\boldsymbol{L}_t;\ \boldsymbol{W}\right)$ predicts transition indicator x_t
  - RNN weights $\boldsymbol{W}$ can be trained to maximise likelihood...
  - ...or (as here) to minimise mean-squared error

$$\widehat{\boldsymbol{W}}\left(\mathcal{D}_t\right) = \underset{\boldsymbol{W}}{\operatorname{argmin}} \sum_t \left(x_t - x\left(\boldsymbol{L}_t;\ \boldsymbol{W}\right)\right)^2$$

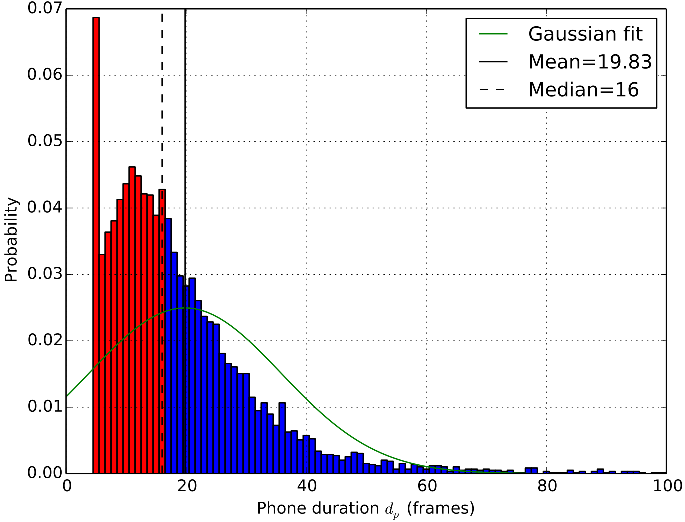
  - Both cases are optimised by the true transition probability $\mathbb{P}\left(X_t = 1 \mid \boldsymbol{L}_t\right)$
- Non-parametric – can describe any duration distribution!
  - Since the NN can give different outputs x for every frame
  - Proper, positive, and possibly skewed, unlike Gaussians
  - Can be run at frame or sample level

From distribution to duration

- Computing the mean of a general non-parametric distribution is not practical
  - Requires an infinite number of π_t-evaluations
- Tail probabilities can be computed from the left tail of the duration distribution only
  - Idea: Perform generation using *quantiles*, points where the tail probabilities reach a certain value q

Quantiles are areas

q-quantile $\widehat{d}(q)$ is the point where red area $\mathbb{P}\left(D_p \leq \widehat{d}\right)$ equals q



Quantile-based generation

- Mathematical definition

$$\widehat{d}_p(q) = \min_{n_t} n_t \text{ such that } q \leq \mathbb{P}\left(D_p \leq n_t\right)$$

where

$$\mathbb{P}\left(D_p > n_t \mid \boldsymbol{L}_t\right) = 1 - \prod_{t'=t_0+1}^{t_0+n_t} \left(1 - \pi_{t'}\right)$$

- Allows sequential generation with no lookahead
- Choosing $q = 1/2$ gives *median-based generation*
  - Median is more probable (typical) than mean, due to skewness

Adding external memory

- To express arbitrary distributions, predictor must be capable of distinct predictions at every frame
  - Possible with an RNN $x\left(\boldsymbol{L}_t;\ \boldsymbol{W}\right)$ due to its internal state
  - Not possible with a DNN $x\left(\boldsymbol{l}_t;\ \boldsymbol{W}\right)$ since $\boldsymbol{l}_t = \boldsymbol{l}_{p(t)}$ is piecewise constant
- Extension: Add a frame counter to the input features $\boldsymbol{l}'_t = [\boldsymbol{l}_t^\intercal\ n_t]^\intercal$
  - RNN $x\left(\boldsymbol{L}'_t;\ \boldsymbol{W}\right)$ no longer have to learn to track n_t
  - DNN $x\left(\boldsymbol{l}'_t;\ \boldsymbol{W}\right)$ now capable of predicting arbitrary distributions
    - Since $\boldsymbol{l}'_t$ changes with every frame

Outline

Experiment setup

- Blizzard Challenge 2016 data (Children's audiobooks)
  - 4.3 hours of data, 4% ($\approx$10 min) for testing
- Feature extraction and code from Merlin (Wu et al., 2016)
- We consider phone duration prediction only
  - No sub-phone states
  - No acoustic model/synthesis yet

Systems

1. Two baselines trained on phone-level data $\mathcal{D}_p$

   Phone-DNN Feedforward DNN

   Phone-LSTM Unidirectional simplified LSTM (Wu and King, 2016)

2. Two proposed systems trained on frame-level data $\mathcal{D}_t$

   Frame-LSTM-I Unidirectional simplified LTSM without...

   Frame-LSTM-E ...or with an external frame-counter input n_t

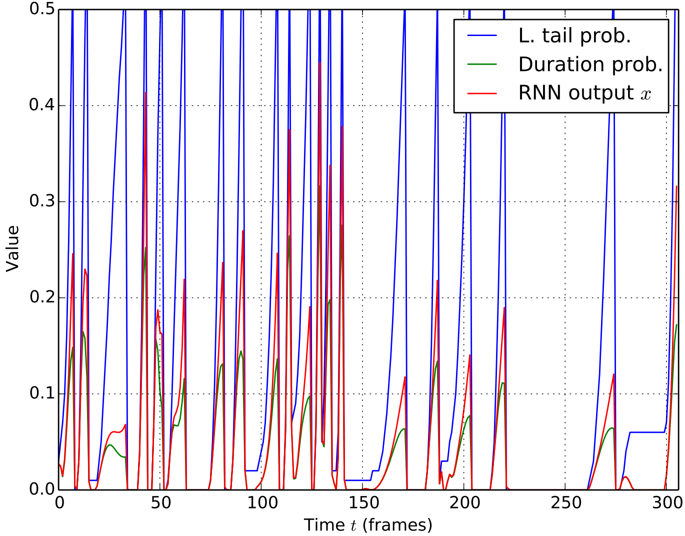
- All used 5 hidden layers of 1024 tanh units each
  - Output layers had 512 (LSTM) or 1024 (DNN) linear units

Training and evaluation

- Learning rate was manually tuned for each system
  - Maximum 25 epochs, with early stopping
- Several evaluation metrics w.r.t. forced-alignment:
  - Root-mean-squared-error (RMSE)
    - Minimised by true mean
  - Mean-absolute-error (MAE)
    - Minimised by true median
  - Pearson correlation (Corr.)
    - Similar to RMSE, but higher is better

Example output

Frame-LSTM-E x in red, $\mathbb{P}(D_p \leq n_t)$ in blue, $\mathbb{P}(D_p = n_t)$ in green



Results

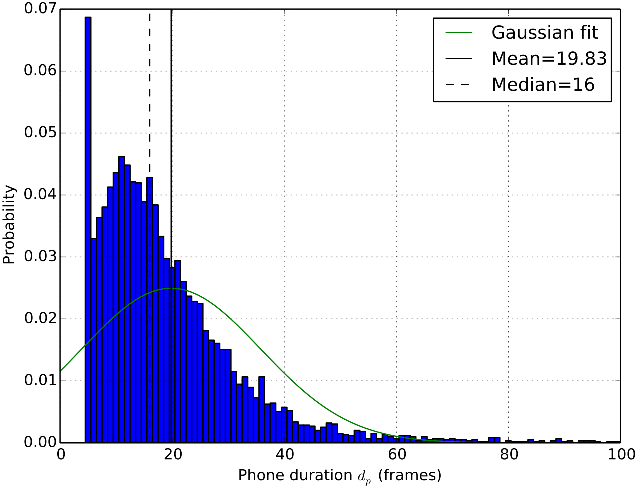
Model	RMSE	MAE	Corr.
Phone-DNN	8.037	4.759	0.750
Phone-LSTM	7.789	**4.556**	0.765
Frame-LSTM-I	8.254	4.610	0.761
Frame-LSTM-E	8.294	**4.574**	0.754

- In MAE, Frame-LSTM-E beats Phone-DNN and is competitive with Phone-LSTM
  - Frame-LSTM-E is worse on vowels, but outdoes Phone-LSTM on all consonant classes except plosives
  - RMSE and correlation are less relevant, since these are not our targets

Outline

Fast speech

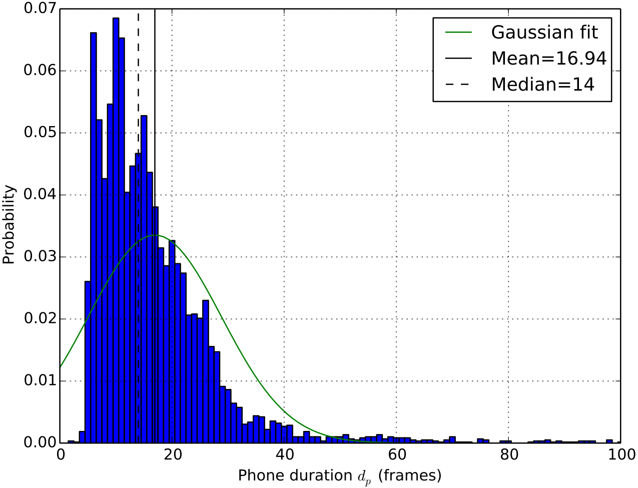
Output durations shorter than data average, due to skewness



(Natural speech)

Fast speech

Output durations shorter than data average, due to skewness



(Median-based generation)

Matching the speaking rate

- We can set the generated quantile $q \neq 1/2$ to alter the speaking rate
- Choose $\widehat{q}$ such that actual and generated mean phone duration match on $\mathcal{D}_p$
  - This $\widehat{q}$ must satisfy

$$\begin{aligned}\overline{d} &\equiv \frac{1}{|\mathcal{D}_p|} \sum_{p \in \mathcal{D}_p} d_p \\ &= \frac{1}{|\mathcal{D}_p|} \sum_{p \in \mathcal{D}_p} \widehat{d}_p(\widehat{q})\end{aligned}$$

  - Same idea can be used to enforce a specific utterance duration

A simple approximation

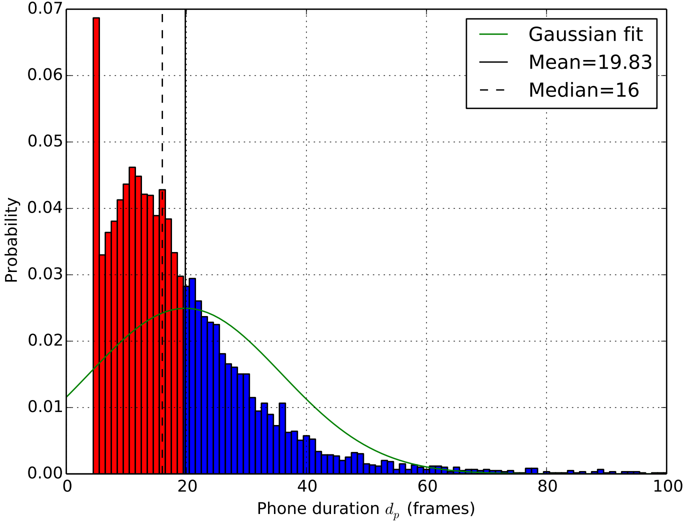
- Finding $\widehat{q}$ requires iteration (e.g., secant method)
- Initialisation/rule of thumb $\widetilde{q}$ based on global duration distribution

$$\widetilde{q} = \frac{1}{|\mathcal{D}_p|} \sum_{p \in \mathcal{D}_p} \mathbb{I}\left(d_p \leq \overline{d}\right)$$

  - Can be computed prior to training

Graphical demonstration

$\widetilde{q}$ is the fraction of the area (red) that is to the left of $\overline{d} = 19.8$



Better aligned speech

- Our joint models define emission and transition probabilities
  - Proper HSMM, but without parametric assumptions
  - HSMM theory and algorithms are directly applicable
- Realignment using NNs can significantly improve TTS quality (Tokuda et al., 2016)
- Fast, local refinements of alignment possible if using a DNN
  - An RNN can then be trained on improved alignments

Local refinement

- Recompute training-data alignments using Viterbi algorithm
  - Constraint: Only allow phone boundaries to move $\pm N$ frames
  - Essentially dynamic time warping on a $(2N + 1) \times |S|$ matrix
- Computational burden is $\mathcal{O}(N|S|)$
  - Linear, not quadratic, in the number of states, $|S|$
- Can be iterated until stable
- Can be done every (few) epoch(s)
- Similar ideas allow most-likely duration generation with fixed global duration

Summary

- We have proposed
  1. Training RNNs/DNNs to predict transition probabilities
  2. Using duration quantiles (e.g., the median) for output generation
- This can describe any duration distribution
- Predicted durations match baseline MAE
- Synthesis, speaking-rate, and realignment are future work

The end

The end

Thank you for listening!

References I

Campbell, W. N. (1989).
Syllable-level duration determination.
In *Proc. Eurospeech*, pages 2698–2701.

Henter, G. E., Ronanki, S., Watts, O., Wester, M., Wu, Z., and King, S. (2016).
Robust TTS duration modelling using DNNs.
In *Proc. ICASSP*, volume 41, pages 5130–5134.

Huber, K. (1990).
A statistical model of duration control for speech synthesis.
In *Proc. EUSIPCO*, pages 1127–1130.

Tokuda, K., Hashimoto, K., Oura, K., and Nankaku, Y. (2016).
Temporal modeling in neural network based statistical parametric speech synthesis.
In *Proc. SSW*, volume 9, pages 113–118.

Watts, O., Henter, G. E., Merritt, T., Wu, Z., and King, S. (2016).
From HMMs to DNNs: where do the improvements come from?
In *Proc. ICASSP*, volume 41, pages 5505–5509.

References II

Wu, Z. and King, S. (2016).
Investigating gated recurrent networks for speech synthesis.
In *Proc. ICASSP*, pages 5140–5144.

Wu, Z., Watts, O., and King, S. (2016).
Merlin: An open source neural network speech synthesis system.
In *Proc. SSW*, volume 9, pages 218–223.

Zen, H., Nose, T., Yamagishi, J., Sako, S., Masuko, T., Black, A., and Tokuda, K. (2007).
The HMM-based speech synthesis system (HTS) version 2.0.
In *Proc. SSW*, pages 294–299.

Zen, H., Tokuda, K., Masuko, T., Kobayashi, T., and Kitamura, T. (2004).
Hidden semi-Markov model based speech synthesis.
In *Proc. Interspeech*, pages 1393–1396.

Preliminary TTS experiment

- Try a system with joint frame-level acoustic-duration model
  - Did not improve perceived speech naturalness over Merlin baseline
  - Not reported in paper (out of space)
- Caveats:
  - Baseline (two NNs) had significantly more parameters
  - Learning rate only tuned for baseline
    - Experiment preceded the reported duration prediction experiment
  - Baseline knows in advance when a phone is about to end
    - Such features improved quality in (Watts et al., 2016)
    - Proposed solution: Use remaining mass and previous-frame acoustic output $\boldsymbol{o}_{t-1}$ as extra inputs, similar to the external frame counter n_t