社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

Non-Parametric Duration Modelling for Speech Synthesis with a Joint Model of Acoustics and Duration

Gustav Eje HENTER[†], Srikanth RONANKI[††], Oliver WATTS[††], and Simon KING[††]

† National Institute of Informatics, Hitotsubashi 2–1–2, Chiyoda-ku, Tokyo 101–8430, Japan
†† The University of Edinburgh, EH8 9LW, United Kingdom

Abstract We describe a new approach to duration modelling for statistical parametric speech synthesis, in which a statistical model is trained to output a phone transition probability at each time unit. Unlike conventional duration modelling – which assumes that duration distributions have a particular shape and use the mean of that distribution for synthesis – our approach can in principle model any distribution supported on the positive integers. Generation from this model can be performed in many ways; here we consider output generation based on the median or other quantiles of the predicted duration. Compared to conventional mean durations, the median is more typical (more probable), is robust to training-data irregularities, and enables incremental generation. Furthermore, our approach is consistent with a longer-term goal of modelling durations and acoustic features together. Results indicate that the proposed method is competitive with baseline approaches in approximating the median duration of held-out natural speech. We also discuss extensions that allow iterative realignment and adjusting the global speech rate.

Key words text-to-speech, speech synthesis, duration modelling, non-parametric models, LSTMs

1. Introduction

This report, an extension of our recent paper [1], describes a new approach to the modelling and generation of segmental durations in synthetic speech. Generating appropriate durations is a challenging but vital step in producing natural-sounding synthetic speech. Statistical parametric speech synthesis (SPSS) has recently made swift improvements through the adoption of deep machine-learning techniques [2], [3]. Recurrent models such as LSTMs [4] may be particularly well suited for prosodic sequence modelling problems, since we expect that high-level, long-range dependencies are of importance for the prosodic structure of speech.

Despite the progress, the prosody of synthetic speech (including durations) remains a major shortcoming. A possible contributing factor is that current systems effectively model durations with a Gaussian distribution and generate predictions from its mean. In reality, the number of frames in a speech segment is positive, integer-valued, and has a skewed distribution, meaning that the mean of a fitted Gaussian will not produce predictions that are most typical of the process (i.e., have a high probability). Another possible weakness of conventional approaches is that they generate durations as an initial stage, separate from acoustic feature generation [4]. We consider it desirable to have a single model whose parameters are learned to simultaneously generate both segment durations and the frames of acoustic features within those segments, since such a joint model, e.g., would allow simultaneous adaptation [5] and control [6] of prosodic and phonetic characteristics in a stable and consistent manner. Simultaneously predicting multiple speech properties may also bring about benefits related to multi-task learning [7]. However, one obvious difficulty in implementing a joint model is that segment durations and acoustic observations conventionally are generated on two separate time-scales. Most recently, WaveNet [8] has enabled competitive waveform-level speech synthesis, but still requires an external duration model.

We here present a non-parametric duration modelling approach that, for each time unit (e.g., acoustic frame), predicts a probability of advancing to the next phone in the phonetic sequence. This can describe any duration distribution on the positive integers, removing the necessity to commit to a predetermined distribution such as Gaussian, log-normal, or gamma. Furthermore, as our model operates on the acoustic frame level, it is capable of unifying models of duration and acoustics, jointly predicting acoustic parameters and phone transition probabilities for each frame. Our approach can in principle be integrated with WaveNet for sample-level duration modelling.

2. Background

This section presents a history of duration generation for

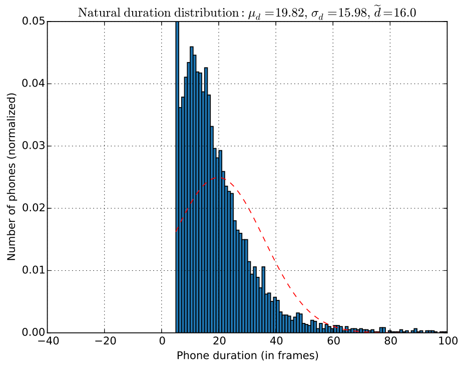

図 1 Histogram of natural durations from our held-out dataset with a fitted Gaussian. The median duration is 16. The minimum duration is five since the HMM-based forced aligner used five no-skip states per phone.

synthesis and describes how our proposal relates to prior work. Theory for our proposed method is provided in Sec. 3.

2. 1 Modelling and Generating Durations

Early, formant-based synthesisers generated phone durations using rules [9], which typically were hand-crafted rather than learned from data. Next followed concatenative synthesisers, which do not require modelling or generating durations, since the units themselves possess intrinsic durations.

Statistical parametric speech synthesis (SPSS) [10], [11] introduced a new duration generation methodology, in which a statistical model (probability distribution) $f_D(d;\,\boldsymbol{\theta})$ is specified for speech-sound durations D. $f_D(d;\,\boldsymbol{\theta})$ is supplemented by i) a machine-learning method predicting properties of f_D (i.e., its parameters $\boldsymbol{\theta}$) from the input text (i.e., the linguistic features $\boldsymbol{l}$), and ii) a principle for generating durations $\widehat{d}$ from the model distribution, such as random sampling or taking the mean of the distribution, $\widehat{d} = \mathbb{E}(D)$.

The first SPSS systems were based on simple hidden Markov models (HMMs), in which state durations $f_D(d;\,\boldsymbol{\theta})$ (usually five per phone) are assumed to follow a memoryless, geometric distribution. Regression trees (RTs) were used to learn the mapping $\boldsymbol{\theta}(\boldsymbol{l})$ based on training data, with the mean of the predicted distribution used for generation.

In reality, natural speech durations, as can be seen in Fig. 1, are not geometrically distributed. Zen et al. [12] introduced the idea of using hidden semi-Markov models (HSMMs) to describe durations in speech synthesis. HSMMs track the amount of time (frames) spent in the current HMM state, and allow the frame counter to influence the probability of s phone transition. This can model a much wider class of duration distributions. Typically, HSMM durations are assumed to follow a parametric family; the widely-used *HMM-based speech synthesis system* (HTS) [13] uses Gaussian distributions, although the skewness and non-negativity of speech durations mean that other distributions might be more suitable, e.g., [14], [15]. As with HMMs, regression trees would be used to predict distribution parameters per state, and the mean of the predicted distribution used for generation.

TTS	Distr. $f_D(d;\,\boldsymbol{\theta})$	Level	Pred. $\boldsymbol{\theta}(\boldsymbol{l})$	Generat.
Formant	-	Phone	-	Rule
Concat.	-	Phone	-	Exemplar
HMM	Geometric	State	RT	Mean
HSMM	Parametric	State	RT	Mean
NN	Gaussian	State	DNN/RNN	Mean
Proposed	Non-parametric	≦Frame	DNN/RNN	Quantile

表 1 Duration generation methods in different TTS types.

Recently, SPSS has seen deep and/or recurrent neural networks (DNNs/RNNs) replace regression trees for learning the mapping $\boldsymbol{\theta}(\boldsymbol{l})$ from text features to duration distributions, e.g., [4], [16]. Typically, a DNN or RNN is trained to minimise mean squared prediction error, which is equivalent to performing mean-based duration generation from a Gaussian model with a globally tied variance.

The evolution of approaches to TTS duration generation is summarised in Table 1. In contrast to all the canonical, prior models in the table, we describe a fundamentally non-parametric approach (“Proposed” in Table 1) where the model is trained to predict the phone transition probability at each time unit. Assuming an asymptotically unbiased learning algorithm, this can in theory describe any distribution on the positive integers, enabling us to represent any and all properties of the duration distribution, for instance the skewness that most previous methods ignore.

2. 2 Quantile-Based Generation

All methods in Table 1 that predict a duration distribution f_D automatically support several duration generation methods based on f_D. In principle, natural durations are random samples from the true duration distribution, but sampling methods have been found to be perceptually unsatisfactory for synthesis unless highly accurate models are used [17]. Consequently, most SPSS systems use deterministic generation methods, which in practice is synonymous with generating the mean duration. We, instead, consider a scheme where synthetic durations are generated from *quantiles* of the predicted duration distribution, such as the median in our experiments. To the best of our knowledge, this is new. Since our distributions need not be symmetric, median durations will typically differ from the distribution mean.

Unlike the mean, quantiles can be identified from the left tail of the distribution. Quantile-based sequential output generation therefore incurs no overhead, which is attractive for incremental synthesis. In addition, our quantiles are always integer-valued. Furthermore, for skewed distributions such as the durations in Fig. 1, the median is frequently closer to the peak of the distribution – the "most typical" outcome – than the mean is; cf. [18]. This follows the spirit of most-likely output parameter generation [19]. Importantly, standard estimates of the median and other quantiles are *statistically robust*, i.e., not unduly affected by the tails of the real duration distribution. Robustness is compelling for TTS [20] as it reduces the sensitivity to unexpected behaviour in the training corpus. This could be of value, e.g., for the highly expressive training data used for the experiments in Sec. 4.

2. 3 High-Resolution Duration Prediction

Frame-level duration predictions are uncommon in the literature, but have a general advantage that they can be unified with the (traditionally distinct) per-frame prediction of acoustic features, such as pitch and vocal-tract filter MGCs. It is suspected that generating durations and fundamental frequency contours that are jointly appropriate may be of importance for synthetic speech prosody; cf. [21], [22].

Jointly modelling durations with acoustic properties of speech was a major motivation for the set-up in [23], where a DNN was trained to output both acoustic parameters and a state-duration vector for each frame. Unfortunately, this necessitates multiple passes during generation and is not easy to interpret probabilistically. Our approach here is probabilistic and can be generalised to sample-level resolution. We are not aware of any other approaches that have considered applicability to sample-level duration generation.

3. Theory

3. 1 Preliminaries

Let $p \in \{1, \ldots, P\}$ be a phone index, and let $t \in \{1, \ldots, T\}$ be an index into timesteps (henceforth frames). (The extension to sub-phone states is straightforward.) Let further D_p – a random variable – be the duration of phone p, and let $d_p \in \mathbb{Z} : d_p > 0$ be an outcome of D_p.

Natural speech phones have duration distributions that depend on the input text. *Duration modelling* in TTS is the task of mapping a sequence of linguistic features $(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_P)$, extracted by the synthesiser front-end, to a sequence of predicted distributions $(D_1, \ldots, D_P)$. *Duration generation*, meanwhile, is the task of mapping $(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_P)$ to a sequence of generated durations $(\widehat{d}_1, \ldots, \widehat{d}_P)$. In SPSS, one or the other of these mappings is learned from parallel text and speech data using machine learning; this report uses RNNs. We write $\mathcal{D}$ to denote a dataset of parallel input features $\boldsymbol{l}$ and duration outcomes d used to train this predictor.

For $q \in (0, 1)$ we say that x is a *q-quantile* of a distribution X if $\mathbb{P}(X \leqq x) = q$. This differs slightly from quantiles in descriptive statistics, which take values on $q \geqq 1$.

3. 2 Conventional Duration Modelling

In statistical synthesisers that generate durations on the state or phone level, the conventional approach is to assume that the durations D_p follow some parametric family of mass functions f_D with a parameter $\boldsymbol{\theta} \in \mathbb{R}^N$, i.e.,

$$\mathbb{P}(D_p = d) = f_D(d;\, \boldsymbol{\theta}_p). \tag{1}$$

Predicting the distribution D_p then reduces to the stochastic regression problem of predicting the distribution parameter $\boldsymbol{\theta}_p$ from the phone-level parallel input-output dataset

$$\mathcal{D}_p = ((\boldsymbol{l}_1, \ldots, \boldsymbol{l}_P),\, (d_1, \ldots, d_P)). \tag{2}$$

We write $\boldsymbol{L}_p$ to denote the linguistic information influencing the predictor at p, which is $\boldsymbol{l}_p$ for feedforward approaches and $(\boldsymbol{l}_1, \ldots, \boldsymbol{l}_p)$ for unidirectional RNNs (on a single utterance).

In practice, most contemporary DNN-based synthesisers do not perform full distribution modelling, but map directly from $\boldsymbol{L}_p$ to a predicted mean $\mathbb{E}(D_p \,|\, \boldsymbol{L}_p)$ of D_p. The dominant principle – also used for the baselines in this study – is to tune the weights $\boldsymbol{W}$ of a DNN or RNN $d(\boldsymbol{L};\, \boldsymbol{W})$ to minimise the training-data mean squared error (MSE),

$$\widehat{\boldsymbol{W}}(\mathcal{D}_p) = \operatorname*{argmin}_{\boldsymbol{W}} \sum_{(\boldsymbol{L}_p,\, d_p) \in \mathcal{D}} (d_p - d(\boldsymbol{L}_p;\, \boldsymbol{W}))^2; \tag{3}$$

synthesis-time durations $\widehat{d_p}$ are then generated by

$$\widehat{d}(\boldsymbol{L}_p) = d(\boldsymbol{L}_p;\, \widehat{\boldsymbol{W}}(\mathcal{D}_p)). \tag{4}$$

The theoretically optimal predictor $\widehat{d}^{\star}$ that minimises the MSE is the conditional mean,

$$\widehat{d}_p^{\star}(\boldsymbol{L}_p) = \operatorname*{argmin}_{\widehat{d}} \mathbb{E}((D_p - \widehat{d})^2 \,|\, \boldsymbol{L}_p) = \mathbb{E}(D_p \,|\, \boldsymbol{L}_p), \tag{5}$$

so the end result is very similar to fitting a Gaussian model f_D and using its mean to generate durations.

3. 3 Non-Parametric Duration Modelling

We now describe a scheme that, unlike conventional parametric families $f_D(d;\, \boldsymbol{\theta})$, is able to model arbitrary frame-level duration distributions for D, by making predictions for each timestep about when phone transitions occur.

Assume phone durations are known up until the current frame t, and let $p(t')$ for $1 \leqq t' \leqq t$ be a function mapping a given frame $t' \leqq t$ to its assigned phone identity. We can then define a frame-level sequence $\boldsymbol{L}_t$ of linguistic features

$$\boldsymbol{L}_t = (\boldsymbol{l}_1, \ldots, \boldsymbol{l}_t) = (\boldsymbol{l}_{p(1)}, \ldots, \boldsymbol{l}_{p(t)}) \tag{6}$$

up until t, with $\boldsymbol{l}_{p(t')}$ constant for all frames in a given phone. For brevity, we write p for the current phone $p(t)$. Finally, we let t_0 be the final frame of the previous phone and define $n_t = t - t_0 \geqq 1$, the duration of the current phone so far.

We now define the *transition probability* π_t for the phone p at time t given $\boldsymbol{L}_t$ – that is, the probability that the current phone p ends on the current frame,

$$\pi_t = \mathbb{P}(D_p = n_t \,|\, D_p \geqq n_t,\, \boldsymbol{L}_t). \tag{7}$$

As long as the transition probabilities satisfy $\pi_t \in [0,\, 1]$ and the infinite product $\prod_{t'=t_0+1}^{\infty}(1-\pi_{t'})$ equals zero they induce a unique, well-defined, positive-integer valued distribution

$$\mathbb{P}(D_p = n_t \,|\, \boldsymbol{L}_t) = \pi_t \prod_{t'=t_0+1}^{t_0+n_t-1} (1 - \pi_{t'}). \tag{8}$$

We propose to build a predictor, based on training data, that estimates π_t from the linguistic input features. Specifically, we will train this predictor on the frame-level dataset

$$\mathcal{D}_t = (\boldsymbol{L}_T,\, (x_1,\, \ldots,\, x_T)), \tag{9}$$

where X_t is an indicator variable that equals one if and only if t is the final frame of the current phone, i.e., iff $t = t_0 + d_p$.

In this report, we consider a deep, unidirectional RNN $x(\boldsymbol{L};\, \boldsymbol{W})$ with weights $\widehat{\boldsymbol{W}}$ trained to minimise the MSE in recursively predicting the indicator variable x_t – that is,

$$\widehat{\boldsymbol{W}}(\mathcal{D}_t) = \operatorname*{argmin}_{\boldsymbol{W}} \sum_t (x_t - x(\boldsymbol{L}_t;\, \boldsymbol{W}))^2. \tag{10}$$

It is easy to prove that the hypothetical predictor $\widehat{x}^{\star}$ that minimises this MSE is

$$\begin{aligned} \widehat{x}^{\star}_t(\boldsymbol{L}_t\,) &= \operatorname*{argmin}_{\widehat{x}} \mathbb{E}\left((X_t - \widehat{x})^2 \,\middle|\, \boldsymbol{L}_t\right) & (11)\\ &= \mathbb{P}(X_t = 1 \,|\, \boldsymbol{L}_t), & (12)\end{aligned}$$

which is mathematically equivalent to π_t. As long as the predictor of X_t is theoretically capable of generating arbitrary, distinct outputs for every frame in a phone, we can describe virtually any transition distribution – and thus any duration distribution. RNNs satisfy this requirement due to their internal state/memory, which evolves with each frame; another solution is described in Sec. 3. 5.

3. 4 From Transitions to Duration Quantiles

Given the predicted duration distribution in (8), we must decide how to generate output durations $\widehat{d}$ from it. As discussed in Sec. 2. 2, sampling typically yields poor naturalness, while mean-based generation is non-robust and unsuitable for sequential synthesis. However, it is easy to compute the right tail probability of the duration distribution through

$$\mathbb{P}(D_p > n_t \,|\, \boldsymbol{L}_t) = \prod_{t'=t_0+1}^{t_0+n_t} (1 - \pi_{t'}). \tag{13}$$

This relation enables synthesis based on *quantiles* $q \in (0,\, 1)$ of the predicted duration. By stepping from $n_t = 1$ and upwards, the (estimated) q-fraction duration $\widehat{d}_p(q)$ of phone p is reached when $\mathbb{P}(D > d)$ first dips down to $1 - q$ or below,

$$\widehat{d}_p(q) = \min_{n_t \in \mathbb{Z}} n_t \quad \text{such that } \mathbb{P}(D_p > n_t) \leqq 1 - q. \tag{14}$$

The median duration is found by setting $q = 1/2$. Eq. (13) thus enables incremental frame generation, advancing $p(t{+}1)$ to the next phone $p+1$ when the desired duration quantile is reached with no additional overhead.

Just as the mean squared error is minimised by the mean, the median is the theoretical minimiser of another error measure, namely the *mean absolute error* (MAE),

$$\mathrm{MAE}(\widehat{d}) = \sum_{p \in \mathcal{D}_p} |d_p - \widehat{d}_p(q)|. \tag{15}$$

A method that improves the MAE might be expected to yield predictions closer to the (conditional) median of the data.

3. 5 Adding a Frame Counter

In the proposal so far, progress through the current phone is tracked solely via the internal state of a recurrent predictor. This contrasts with hidden semi-Markov models, which achieve arbitrary (parametric) duration distributions by maintaining an external counter of the number of frames spent in each state, from which transition probabilities are computed. However, nothing prevents adding that variable, n_t, as an input to a neural network $x(\,\cdot\,;\, \boldsymbol{W})$ that predicts π_t, in addition to the regular, linguistic features $\boldsymbol{L}_t$. We call these augmented features $\boldsymbol{l}'_t$ and

$$\boldsymbol{L}'_t = ([\boldsymbol{l}_1^{\intercal},\, n_1]^{\intercal},\, \ldots,\, [\boldsymbol{l}_t^{\intercal},\, n_t]^{\intercal}), \tag{16}$$

so that the augmented RNN predictor is $x(\boldsymbol{L}';\, \boldsymbol{W})$.

Since n_t is highly relevant for duration prediction, providing it as an explicit input feature is likely to increase performance over relying on capturing the same information only implicitly. To test this hypothesis, our experiments in Sec. 4. compare two systems that differ only in whether or not they include n_t as an input to the frame-level RNN.

As a side note, the frame-level inputs $\boldsymbol{l}'_t$ differ with each frame. This makes it possible for predicted transition distributions to vary from frame to frame as well, even without using stateful predictors such as RNNs. In principle, this could allow feedforward DNNs using $\boldsymbol{l}'_t$ to also express arbitrary duration distributions, though we have not explored this possibility in the experiments reported here.

4. Experimental Validation

We here recount an experiment previously reported in [1].

4. 1 Data

For an initial evaluation of frame-level duration prediction we used the speech database from the 2016 Blizzard Challenge [24], comprising speech and text of 50 children's audiobooks read by a British female speaker. The total audio duration was about 4.33 hours after segmentation. 4% of the data (three whole stories) were set aside as a test set.

4. 2 Feature Extraction

A state-level forced alignment of the segmented data was obtained using context-independent HMMs. To improve the accuracy of forced-aligned durations, `ehmm` [25] was used to insert pauses into the annotation based on the acoustics. The remaining feature extraction was very similar to that in Merlin [26], with 481 text-derived, normalised binary and numerical features used for linguistic features $\boldsymbol{l}$. The number of frames in each phone was used as the prediction target for the conventional baselines. Unlike [20], sub-phone states were not used in either training or prediction.

4. 3 Training and Synthesis

Four systems were trained: two conventional phone-level predictors (*Phone-DNN* and *Phone-LSTM*) and two proposed systems (*Frame-LSTM-I* and *Frame-LSTM-E*). Each network was initialised with small, random weights, and then trained for 25 epochs with a manually tuned learning rate. Validation-set early stopping was used to avoid overfitting.

Both baselines used phone-level linguistic features as inputs and were optimised to minimise the MSE duration-prediction error. *Phone-DNN* was a feedforward DNN with six layers of 1024 nodes each. *Phone-LSTM* was configured with five feedforward layers of 1024 nodes each and a final uni-directional SLSTM [27] hidden layer of 512 nodes.

The two proposed systems used the same architecture as that of *Phone-LSTM* but were trained with 1 datapoint per frame instead of per phone. Their prediction targets were 0.0 for non-phone-final frames and 1.0 for phone-final frames. *Frame-LSTM-I* used $\boldsymbol{l}_t$ as input, forcing it to rely on *internal* memory (RNN state) for frame counting. *Frame-LSTM-E* used inputs $\boldsymbol{l}'_t$ with an external frame counter as in Sec. 3. 5.

Synthesis was performed from phone sequences with an oracle pausing strategy (pauses inserted by `ehmm` based on test data). For *Phone-DNN* and *Phone-LSTM*, predicted (mean) durations were rounded to whole frames. *Frame-LSTM-I* and *Frame-LSTM-E*, meanwhile, used the frame-wise duration generation technique from Sec. 3. 4 to generate approximate median ($q = 1/2$) durations.

4. 4 Results

Table 2 presents RMSE, MAE (both in frames per phone), and Pearson correlation between predicted durations and the held-out reference, ignoring silences. It is obvious that the RNN (*Phone-LSTM*) is superior to the feedforward DNN (*Phone-DNN*) for duration prediction. For the proposed systems, the mean-based baselines outperform the proposed methods in terms of RMSE and Pearson correlation (mathematically very similar to the RMSE), but that gap is much smaller when it comes to MAE. This pattern is expected, since, as explained in Sec. 3. 4, the median is the theoretical minimiser of MAE while the mean minimises the (R)MSE.

Table 3 shows experimental results broken down by phonetic class. Interestingly, while *Frame-LSTM-E* has worse MAE than the LSTM benchmark for vowels and slightly worse for consonants overall, for all consonant classes except plosives the proposed method performs better. While *Frame-LSTM-E* does not surpass the baseline even on MAE, the performance gap is effectively closed (4.556 vs. 4.574). We have thus devised a system with similar MAE as existing predictors, but with greater compatibility with our acoustic models as it, too, operates on a frame level.

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

表 2 Objective metrics for predicted durations measured w.r.t. forced-aligned durations.

Phonetic	Phone-LSTM			Frame-LSTM-E		
class	RMSE	MAE	Corr.	RMSE	MAE	Corr.
Vowel	8.516	**4.848**	0.809	9.027	4.891	0.799
Consonant	7.313	**4.378**	0.709	7.815	4.382	0.694
Plosive	5.206	**3.608**	0.732	5.610	3.612	0.720
Fricative	6.489	4.380	0.769	6.859	**4.246**	0.764
Nasal	6.833	4.459	0.568	7.376	**4.398**	0.550
Affricate	5.658	4.220	0.797	5.432	**3.746**	0.821
Glide/liquid	8.013	5.260	0.569	8.235	**5.075**	0.599

表 3 Objective measures broken down by phonetic class.

5. Extensions

Modelling transition probabilities enables several extensions of conventional synthesis, as outlined in this section.

5. 1 Tuning the Speaking Rate

Deterministic output-generation methods, e.g., using means or medians, need not produce output that matches the training data in all aspects – cf. the global variance of mean-based generation [28, Sec. 6.2.3 & Sec. 6.5.2]. Real duration distributions are skewed, so median-based generation leads to sped-up speech (shorter average phone duration) compared to the training data. If this is inappropriate, quantile-based generation allows tuning the overall speaking rate by adjusting q. We can choose $\widehat{q}$ to make the average actual and generated phone durations match over $\mathcal{D}_p$ by solving

$$\overline{d} \equiv \frac{1}{|\mathcal{D}_p|} \sum_{p \in \mathcal{D}_p} d_p = \frac{1}{|\mathcal{D}_p|} \sum_{p \in \mathcal{D}_p} \widehat{d}_p(\widehat{q}) \tag{17}$$

for $\widehat{q}$. The resulting value typically exceeds $1/2$.

Eq. (17) cannot be solved analytically, but one can use iterative root-finding schemes to identify a proper $\widehat{q}$ as a final stage of model training. Iterations may be initialised from a starting q-value $\widetilde{q}$ based on the relation between distribution quantiles and the mean duration in the training data, as in

$$\widetilde{q} = \frac{1}{|\mathcal{D}_p|} \sum_{p \in \mathcal{D}_p \,:\, d_p \leqq \overline{d}} 1. \tag{18}$$

$\widetilde{q}$ can be computed prior to training using only the global duration distribution graphed in Fig. 1, and may provide a first approximation $\widetilde{q} \approx \widehat{q}$ even without iterative root-finding.

5. 2 Realigning the Training Data

Another advantage of transition probability modelling is that it maps neatly onto the classical theory of HSMMs (though our approach is substantially more powerful than, e.g., [13] or [29]). We can thus use techniques from HMMs or HSMMs to analyse and extend our approach. It is for example compelling to use the Viterbi algorithm to locally refine(注1) alignments using a DNN-based synthesiser with durations predicted as in Sec. 3. 5. Better alignments may benefit both durations and acoustics: a study on Gaussian HSMMs using DNN predictors for realignment [29] found substantial quality increases over a baseline that did not realign. DNN-refined alignments may further be used to train RNN TTS.

6. Conclusion

We described a new duration-modelling paradigm with DNNs/RNNs that predict phone or state transition probabilities in sync with the acoustic model in a speech synthesiser. The next step is to subjectively evaluate joint modelling of duration and acoustics, with or without constraining the speaking rate to match the training-data speech rate.

文献

[1] S. Ronanki, O. Watts, S. King, and G. E. Henter, “Median-based generation of synthetic speech durations using a non-parametric approach,” in *Proc. SLT*, vol. 6, 2016.

[2] H. Zen, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *Proc. ICASSP*, 2013, pp. 7962–7966.

[3] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, “Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis,” in *Proc. ICASSP*, 2015, pp. 4460–4464.

[4] H. Zen and H. Sak, “Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis,” in *Proc. ICASSP*, 2015, pp. 4470–4474.

[5] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, “A study of speaker adaptation for DNN-based speech synthesis,” in *Proc. Interspeech*, vol. 16, 2015, pp. 879–883.

[6] O. Watts, Z. Wu, and S. King, “Sentence-level control vectors for deep neural network speech synthesis,” in *Proc. Interspeech*, 2015, pp. 2217–2221.

[7] R. Caruana, “Multi-task learning,” *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[8] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “WaveNet: A generative model for raw audio,” *arXiv preprint 1609.03499*, 2016.

[9] D. H. Klatt, “Review of text-to-speech conversion for English,” *J. Acoust. Soc. Am.*, vol. 82, no. 3, pp. 737–793, 1987.

[10] H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[11] S. King, “An introduction to statistical parametric speech synthesis,” *Sadhana*, vol. 36, no. 5, pp. 837–852, 2011.

[12] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Hidden semi-Markov model based speech synthesis,” in *Proc. Interspeech*, 2004, pp. 1393–1396.

[13] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, “The HMM-based speech synthesis system (HTS) version 2.0,” in *Proc. SSW*, 2007, pp. 294–299.

[14] W. N. Campbell, “Syllable-level duration determination,” in *Proc. Eurospeech*, 1989, pp. 2698–2701.

[15] K. Huber, “A statistical model of duration control for speech synthesis,” in *Proc. EUSIPCO*, 1990, pp. 1127–1130.

[16] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, “Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices,” in *Proc. Interspeech*, 2016, pp. 2273–2277.

[17] G. E. Henter, T. Merritt, M. Shannon, C. Mayo, and S. King, “Measuring the perceptual effects of modelling assumptions in speech synthesis using stimuli constructed from repeated natural speech,” in *Proc. Interspeech*, 2014, pp. 1504–1508.

[18] H. L. MacGillivray, “The mean, median, mode inequality and skewness for a class of densities,” *Aust. J. Stat.*, vol. 23, no. 2, pp. 247–250, 1981.

[19] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, “Speech parameter generation algorithms for HMM-based speech synthesis,” in *Proc. ICASSP*, vol. 25, 2000, pp. 1315–1318.

[20] G. E. Henter, S. Ronanki, O. Watts, M. Wester, Z. Wu, and S. King, “Robust TTS duration modelling using DNNs,” in *Proc. ICASSP*, vol. 41, 2016, pp. 5130–5134.

[21] R. Fernandez, A. Rendel, B. Ramabhadran, and R. Hoory, “Prosody contour prediction with long short-term memory, bi-directional, deep recurrent neural networks,” in *Proc. Interspeech*, 2014, pp. 2268–2272.

[22] S. Ronanki, G. E. Henter, Z. Wu, and S. King, “A template-based approach for speech synthesis intonation generation using LSTMs,” in *Proc. Interspeech*, 2016, pp. 2463–2467.

[23] O. Watts, S. Ronanki, Z. Wu, T. Raitio, and A. Suni, “The NST–GlottHMM entry to the Blizzard Challenge 2015,” in *Proc. Blizzard Challenge Workshop*, 2015.

[24] S. King and V. Karaiskos, “The Blizzard Challenge 2016,” in *Proc. Blizzard Challenge Workshop*, 2016.

[25] K. Prahallad, A. W. Black, and R. Mosur, “Sub-phonetic modeling for capturing pronunciation variations for conversational speech synthesis,” in *Proc. ICASSP*, 2006, pp. I–853–I–856.

[26] Z. Wu, O. Watts, and S. King, “Merlin: An open source neural network speech synthesis system,” in *Proc. SSW*, vol. 9, 2016, pp. 218–223.

[27] Z. Wu and S. King, “Investigating gated recurrent networks for speech synthesis,” in *Proc. ICASSP*, 2016, pp. 5140–5144.

[28] M. Shannon, “Probabilistic acoustic modelling for parametric speech synthesis,” Ph.D. dissertation, Department of Engineering, University of Cambridge, Cambridge, UK, 2014.

[29] K. Tokuda, K. Hashimoto, K. Oura, and Y. Nankaku, “Temporal modeling in neural network based statistical parametric speech synthesis,” in *Proc. SSW*, vol. 9, 2016, pp. 113–118.

(注1) : By *local refinment*, we mean optimal Viterbi alignment with each phone boundary constrained to move less than a certain number of timesteps from its current location. This has a complexity linear in the number of phones. If boundaries change, the utterance can be realigned again. Realignment can be performed every few epochs.