10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

Where do the improvements come from in sequence-to-sequence neural TTS?

Oliver Watts[1], *Gustav Eje Henter*[2], *Jason Fong*[1], *Cassia Valentini-Botinhao*[1]

[1]The Centre for Speech Technology Research, The University of Edinburgh, Edinburgh, UK
[2]Division of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden
owatts@inf.ed.ac.uk

Abstract

Sequence-to-sequence neural networks with attention mechanisms have recently been widely adopted for text-to-speech. Compared with older, more modular statistical parametric synthesis systems, sequence-to-sequence systems feature three prominent innovations: 1) They replace substantial parts of traditional fixed front-end processing pipelines (like Festival's) with learned text analysis; 2) They jointly learn to align text and speech and to synthesise speech audio from text; 3) They operate autoregressively on previously-generated acoustics. Naturalness improvements have been reported relative to earlier systems which do not contain these innovations. It would be useful to know how much each of the various innovations contribute to the improved performance. We here propose one way of associating the separately-learned components of a representative older modular system, specifically Merlin, with the different sub-networks within recent neural sequence-to-sequence architectures, specifically Tacotron 2 and DCTTS. This allows us to swap in and out various components and subnets to produce intermediate systems that step between the two paradigms; subjective evaluation of these systems then allows us to isolate the perceptual effects of the various innovations. We report on the design, evaluation, and findings of such an experiment.

Index Terms: Speech synthesis, end-to-end, SPSS, naturalness

1. Introduction

In recent years, attention-based sequence-to-sequence (seq2seq) neural networks have been widely adopted for text-to-speech (TTS) synthesis [1, 2, 3]. Various motivations for using such models have been advanced. Most generally, use of these models leads to improved naturalness of synthetic speech compared with that generated by previous-generation statistical parametric speech synthesis (SPSS). This paper seeks to determine which elements of the new paradigm contribute most to these improvements in naturalness.

SPSS and seq2seq systems differ in numerous ways. There are however three prominent differences between the paradigms (detailed in Sec. 2) which we believe may account for the superior performance of seq2seq. First, new systems replace large parts of traditional hand-engineered text-analysis pipelines (like Festival's) with learned neural text encoders. Second, alignments – the association between linguistic symbols and frames of speech in the training data – are learned jointly with the synthesis model, instead of being determined ahead of time by, e.g., HMM-based forced alignment, and then fixed during acoustic model training. Thirdly, seq2seq models operate autoregressively on the acoustics generated at previous time steps, instead of predicting adjacent acoustic frames independently given the labels or hidden unit activations underlying them.

We here seek to determine the extent of each of the three factors' contribution to improvements in synthetic speech naturalness. Our motivation and methodology is similar to that of earlier work [4], in which we determined what aspects of a then-standard deep neural network (DNN) based SPSS system contributed to its superior performance relative to one based on HMMs and decision trees, by designing a range of systems which allowed us to isolate the effect of various innovations. Designing a similar experiment to compare SPSS and seq2seq TTS is more complicated, however. For instance, there are non-trivial dependencies between the three factors described. One example is autoregression and attention: it is possible to have an autoregressive system without attention, but attention entails some form of autoregression. This and other interactions between the factors mean that it is not possible to build systems which correspond to all combinations of factors. Instead, we design a range of systems that allows us to step gradually from the older paradigm to the newer. This can be done in more than one way, and is complicated by the structural differences between the systems – the fact that the subnets of seq2seq systems do not unambiguously map onto the clearly-delineated modules in SPSS. To step from one paradigm to the other, we must first propose a tentative functional mapping between seq2seq subnets and SPSS modules, and then decide on an order in which the functional blocks are swapped from old to new to produce intermediate systems. Table 1 attempts to provide an overview of the major differences between a prominent SPSS system (Merlin [5]) and two representative neural seq2seq systems (Tacotron 2 [2] and DCTTS [3]), demonstrating one possible functional mapping between systems and paradigms. In this work, we describe an investigation stepping from Merlin to DCTTS to measure the effect of differences in design between the two systems, though we believe our findings generalise beyond these specific implementations.

Any given pair of SPSS and seq2seq systems will exhibit additional differences beyond the three factors we have highlighted, for instance in the acoustic feature extraction and/or details of the optimisation used. Even if the three principal factors outlined above might be the most important inter-paradigm differences, there is reason to suspect that other differences are the consequence of carefully tuning each approach for best results. When combined, these smaller differences make it hard to compare an older-paradigm system with a new paradigm system directly, and a careful experiment must control for their influence in order to draw meaningful conclusions.

New architectures have other advantages beyond improving speech naturalness. For example, using attention simplifies voice creation by fusing multiple steps (alignment, duration and acoustic model training) into one. While other such motivations are important, we here focus only on the effect of system design on speech naturalness ratings.

2. From SPSS to Seq2Seq

In this section we give a high-level survey of the three principal inter-paradigm differences of interest in our study; lower-level

10.21437/SSW.2019-39

Table 1: *One possible mapping between SPSS (Merlin) and seq2seq neural TTS (Tacotron 2 and DCTTS) components and interfaces.*

	Step or property	Merlin [5]	Tacotron 2 [2]	DCTTS [3]
Front end	Text pre-processing	Text norm. → phon. dict. + G2P for OOV	Text norm. → (optional) phon. dict. + G2P for OOV	Text norm. → (optional) phon. dict. + G2P for OOV
	Linguistic analysis	Festival [6] (part-of-speech, position in utt., etc. → punctuation removal)	Encoder subnet (CNN and BD-LSTM)	TextEnc subnet (CNN and highway)
Align & dur.	Alignment	Monophone forced alignment from HTS [7]	Location-sensitive attention [8]	Guided attention [3]
	Aligner acoust. feats.	Low-order MGCs (used by HTS)	Decoder attention LSTM hidden state	AudioEnc (CNN and highway) output Q
	Duration prediction	Separate RNN using only text-derived feats.	Location-sensitive attention [3]	Forcibly incremental attention
Acoustic model	Frame-level progress indicator	State & phone pos. and dur. feats.	Pre-net (feed-forward DNN) output and LSTM hidden state	AudioEnc output Q
	Acoustic predictor	Acoustic model (feed-forward DNN)	Decoder subnet (LSTM and linear)	AudioDec subnet (CNN and highway)
	Loss function	MSE (L2)	MSE (L2) + post-net MSE + stop token binary cross-entropy (BCE)	L1 + BCE on acoustics + guided attention loss (not in SSRN)
Waveform generation	Acoustic pred. target feats.	WORLD [9] VUV, $\log F_0$, MGCs, BAPs + dynamic feats. → mean & var. norm.	Log power mel-spec. clipped to a min. power	Pre-emphasis → mel power spec. → max & min norm. → $y = x^\gamma$ transformation
	Acoustic feat. dim. & time resolution	$1 + (1 + 60 + 1) \times 3 = 187$ every 5 ms	80 every 12.5 ms (before and after post-net)	80 every 50 ms (1025 lin. spec. every 12.5 ms after SSRN)
	Acoustic feat. post-proc.	MLPG [10] → postfilter → cep2spec	Post-net (non-causal CNN)	SSRN (non-causal CNN and highway)
	Time-domain conversion & proc.	WORLD synthesis	Mixture-of-logistics WaveNet [2] conditioned on post-net mel-spec.	Griffin-Lim [11] phase recovery → undoing pre-emphasis
	Separately learned components	HTS forced-aligner; duration model; acoustic model	Spectrogram pred. net; WaveNet (w/ ground-truth or pred. spectra)	Text2Mel network; SSRN

implementation details are left for Sec. 3 below.

2.1. Fixed linguistic analysis vs. learned text encoding

Most work on 'end-to-end' TTS in fact assumes a fixed, manually-specified text-normalisation module to preprocess incoming text [12]. Furthermore, rather than learning to operate directly on characters it is commonplace to first convert plain text to sequences of phones and use these as inputs to the learned modules [13]. We denote this initial treatment of text as *text preprocessing* in the first row of Table 1. It can be seen that this stage is the same for systems of the old and new paradigms if phone input is used.

The first general difference happens in what we term *linguistic analysis*: in SPSS, phone inputs are supplemented with many hand-engineered, language-specific linguistic features, relating, e.g., to phonetic classes, position in units such as syllable, word and phrase, syntactic categories of words, and detected or predicted pitch accents and phrase-boundaries.

For current purposes we assume that anything that happens in a seq2seq model before attention is computed corresponds to the linguistic-analysis module of a modular system, and that the subnets which map from attention-derived contexts to audio correspond to a conventional acoustic model. That this division is approximate is made especially clear in systems which use dot-product attention [14], and where a text encoder therefore is required to create representations in the same space as ones derived from acoustics by audio-encoders/pre-nets [3]. This identification is therefore somewhat tentative: the very nature of jointly-optimised subnets means that this must be the case.

2.2. Fixed phonetic alignment vs. attention mechanism

In older SPSS systems, an alignment of phone symbols to frames of training data is carried out before and then kept fixed during acoustic and duration model training. The Merlin system [5] – which uses 5-state context-independent phone HMMs over frames of speech at 5 ms intervals – is representative of such older systems. Many modern systems, in contrast, use an *attention mechanism* – made popular by machine translation – to jointly align linguistic symbols with frames of speech and learn the mapping from one to the other [1].

In principle, attention mechanisms could incorporate textual cues distributed across the whole length of an input sequence to condition the output at any timestep. In more general formulations of attention there is no encouragement for the association of inputs and outputs to be monotonic; indeed, such monotonicity would be incompatible with the requirements of machine translation, where reordering of word sequences between languages is common. At first sight, it might be supposed that TTS systems make use of attention to model long-range contextual dependencies as well as to align sequences of different lengths and varying relative rates. However, inspection of attention plots from TTS systems published by others and created in our own work suggests that this is not the case, and TTS systems mainly learn to employ attention in order to capture local phonetic-acoustic alignments. Most values in these plots are near 1 or 0, with some intermediate values at phonetic segment boundaries, and the progress of alignment is always monotonic. Monotonicity can be encouraged or enforced by explicit constraints, but we have observed the same tendency also when training systems with unrestricted attention. Long-distance dependencies thus seems to be resolved by text-encoder modules, and we confidently identify the attention mechanism as a drop-in replacement for forced alignment (in training) and the duration model (at synthesis time).

2.3. Subphone positional features vs. acoustic feedback

We identify the nine subphone positional features in [5], which are appended to phone-level features (which consist of normalised counts such as *Fraction through state counting forwards/backwards*), with the acoustic feedback used in autore-

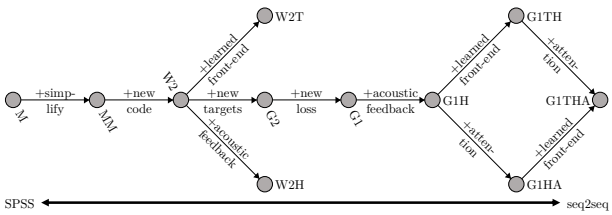

Figure 1: *Schematic view of relationships between models.*

gressive systems [3]. The reason for this identification is functional: in [5] these extra features serve to distinguish between otherwise-identical frames of linguistic features so that smoothly varying predictions can be made across the course of a phonetic segment. Similarly, it is the use of acoustic history which enables a recurrent decoder to generate smoothly varying output, and to determine when to move the focus of its attention to the next input symbol.

We note that output-recurrences in TTS are not an innovation of the new paradigm, and are employed also by systems which make no use of attention, although we note that in, e.g., [15] the conditioning on previous acoustics is relatively shallow, in that past predictions are incorporated into the current prediction through a single layer of a neural network. In contrast, [2, 3] feed previous acoustics through many layers of prenet/audio-encoder and decoder between adjacent timesteps.

3. Systems built

3.1. Rationale

The differences between the different systems compared in this paper are summarised in Table 2. A graphical overview of the relationships and minimal differences between the systems is provided in Fig. 1. Samples of synthetic speech produced by all systems and a link to the modified codebase used to create them can be found at homepages.inf.ed.ac.uk/owatts/papers/SSW10.

The systems at the left and right extremes of the graph in Fig. 1 – M and G1THA – correspond closely to [5] and [3], respectively. Intermediate systems were built in order to answer the research questions summarised in Table 3. Starting at G1THA and moving gradually leftwards to G1 involves removing the three important characteristics which distinguish systems in the modern paradigm, discussed above in Sec. 2, and denoted in Fig. 1 with the letters T (***t***ext analysis), H (acoustic ***h***istory) and A (***a***lignment). Stepping by different routes involves removing factors in different orders; the combinations we have chosen are all trainable and allow many comparisons to help answer the primary questions in Table 3. Due to the other differences between systems in the old and new paradigms listed in Table 2, G1 is however a very different system from M. Stepping from G1 leftward to W2 brings it closer, by moving to the same L2 loss function employed by M (denoted by **2** in G2), and then to the same WORLD features (denoted by **W** in W2). These extra steps are intended to allow us to answer questions 6 and 7 in Table 3. The side-branches leading to W2T and W2H explore the effect of varying two of the main factors of interest (T and H) while using a loss loss function and acoustic features typical of the older paradigm, and allow further system comparisons to test the primary questions listed in Table 3.

W2 is still distant from M; this is partly because it is implemented with a different codebase, and employs network architectures and hyperparameter settings having more in common with G1THA than M. Furthermore, although WORLD parameters are used to be more consistent with M, the frameshift used for W2 (12.5 ms) is larger than that used by M (5 ms). We therefore constructed system MM, where system M is modified to use the same frameshift as W2 and the same codebase, basic architectures and optimisation as M, and thus is intended to allow us to answer questions 4 and 5 in Table 3.

Questions 8 and 9 in Table 3 relate to interactions between separate system elements which we suppose might have an impact on system performance. We pose question 8 as we suppose that acoustic feedback might give bigger gains in performance when there is fine spectral detail which must be compatible between adjacent generated frames of speech. This is the case with mel-spectrograms, for example, where F0 is not represented explicitly but is encoded in spectral harmonic structure. We suppose that the inconsistency across time due to independent prediction of neighbouring frames would be more harmful in this case than in the case of WORLD features, where F0 is (ideally) encoded independently of spectral envelope.

We pose question 9 in Table 3 because we suspect that the success of attention is reliant on a text encoder with sufficient flexibility to produce representations which are directly comparable with ones derived from acoustics. As noted above, the dot-product attention in [3] assumes such comparability.

3.2. Merlin based systems, M and MM

M is neural-network based SPSS as implemented in Merlin with the standard recipe [5]. It consists of two distinct modules: The fixed 'front end' first uses Festival [6] to predict contextual linguistic features from raw text (of the same format as in HTS [16]), and then uses HTK [17] to force-align these features to vocoder features generated from the ground truth audio. These time-aligned features are then used as inputs and outputs to train Merlin's 'back end', a feed forward network acoustic model that predicts one frame of acoustic features for each frame of linguistic features. These acoustic features are then converted into a waveform using WORLD [9].

The contextual linguistic features are aligned at the HMM-state rather than the phone level, as this allows different segments of the phone (e.g., the closure and burst of a plosive) to be modelled with separate HMM parameters providing a more fine-grained acoustic segmentation to the DNN model, so that it can learn to better predict duration and acoustic features. Both linguistic and acoustic features are generated at a frame rate of 5 ms. The sampling frequency used was 16 kHz. The DNN acoustic model has 6 layers, each of which has 1024 units, all using a $\tanh$ activation function. The model was trained using SGD with an exponentially decaying learning rate starting at 0.002 and batch size 256. It is trained for 25 epochs.

MM adapts **M** to use a 12.5 ms frame rate for linguistic and acoustic features in order to match the frame rate of DCTTS/G1THA. To accommodate this change we reduce the number of states within each phone's HMM alignment model from 5 to 3. From informal listening this improves the quality. This is likely because using 5 states with a 12.5 ms frame rate *cannot* model shorter-duration phones whereas using 5 states with a shorter frame rate of 5 ms *can* (5×12.5 ms $> 5\times5$ ms).

3.3. DCTTS based systems (W and G)

All further systems were based on [18], an approximate implementation of [3]. The code was considerably modified to allow (among other things) the training of between-paradigm systems and the use of phonetised inputs. Hyperparameters for training all systems except M and MM were inherited from this implementation and were not thoroughly optimised for each configur-

Table 2: *Systems compared. The signal generator also determines the acoustic features used (vs. 80-filterbank mel-spectrum log-magnitudes). Learned front-end is DCTTS TextEnc. Fixed alignments are monophone HTS; learned alignments are (guided) attention.*

System	Codebase	Frame hop	Dynamic feats.	Sig. gen.	Acoust. loss	Front-end	Feedback	Alignment	SSRN
M	Merlin	5 ms	$\Delta + \Delta^2$	WORLD	L2	Fixed	As in [5]	Fixed	N/A
MM	"	12.5 ms	"	"	"	"	"	"	"
W2	DCTTS	50 ms	None	"	"	"	Rel. pos. in phone	"	"W2"
W2T	"	"	"	"	"	Learned	"	"	"
W2H	"	"	"	"	"	Fixed	Acoustic	"	"
G2	"	"	"	G-L	"	"	Rel. pos. in phone	"	"G2"
G1	"	"	"	"	L1 + BCE	"	"	"	"G1"
G1H	"	"	"	"	"	"	Acoustic	"	"
G1TH	"	"	"	"	"	Learned	"	"	"
G1HA	"	"	"	"	"	Fixed	"	Learned	"
G1THA	"	"	"	"	"	Learned	"	"	"

ation. We expect the inherited parameters to suit configuration G1THA – the system most similar to DCTTS – best, and systems to the left of G1THA in Fig. 1 progressively worse. This has implications for our results: we have greater confidence that findings from comparisons between systems towards the right-hand side of this scale will be more likely to generalise to situations where hyperparameters are well optimised per system.

All systems described below follow the original formulation of [3] in that they consist of two independently-trained networks; the first maps textual or linguistic units to low resolution acoustic features, and the second (a so-called 'spectrogram super-resolution network', or SSRN) upsamples those acoustics in time and (in some cases) along the frequency axis.

All systems starting with the code W use the same WORLD parameters as system MM, extracted with a 12.5 ms frameshift. The same mel-cepstral representation is used for spectral envelope parameters. The SSRN for these systems (called 'W2' in Table 2) upsamples in time, accepting inputs at 50 ms intervals and outputting at 12.5 ms intervals. It performs no adjustment to the dimensionality of each frame of parameters, in contrast to the mel-spectral SSRNs used for the G systems.

Also, unlike in systems M and MM, none of the SSRNs we trained made use of dynamic features. Dynamic features are required by the M systems for use by MLPG to generate static speech parameter trajectories which vary realistically over time. We consider this to be one of the responsibilities of the SSRN in the W and G systems, hence the absence of these features.

Factor T: Systems W2T, G1TH, and G1THA all make use of learned text analysis; that is, they accept as input phone sequences enriched with punctuation and word-boundary symbols which are fed into a text encoder of the same specification as that described in [3], consisting of several dilated convolutional layers and highway blocks. For the remaining systems, switch I in Fig. 2 is adjusted, and the same phone-level labels used for system M are used in place of the text encoder's output V. In these systems, a single learned linear transform is applied to the labels to convert them to the same dimensionality as V. System G1HA uses the same transformed features also to compute attention (i.e., for both K and V in Fig. 2).

Factor A: Systems G1HA and G1THA make use of learned attention, where elements of matrix A (shown in Fig. 2) are computed as the dot-product between frames of K and Q. As in [3], an extra attention loss is used for optimisation in these cases. In other systems, switch II in Fig. 2 is flipped and A is a binary matrix populated for each sentence from the fixed, HMM-based alignments used by system M (resampled in time to the appropriate frame-rate). As A is binary in these cases (elements either 1 or 0), it acts as a selection matrix: in multiplying V by it, we are effectively selecting context-dependent phone representations from V and repeating each of them the appropriate number of times to match the acoustic frame rate.

Factor H: Systems W2H, G1H, G1HA, G1TH, and G1THA all condition their predictions on previous frames of acoustics. As in [3], ground-truth acoustics are used in training, and previously-generated ones at synthesis time. Note that Q and H in Fig. 2 – which could in principle be different representations – are identical in the systems we built, consistent with [3]. For all other systems, switch III is flipped, and predictions are conditioned on a simple frame-counter feature (normalised position forwards in phone) rather than encoded acoustic history. Our frame-counter is a scalar feature with no state-level alignment information; this is rather less informative than the 9 subphone features employed by M and MM.

3.4. Data

All systems were trained on *LJ Speech* [19], a public-domain single-speaker speech database of approximately 24 hours of transcribed speech read from non-fiction books by a female speaker of American English. This data was chosen as it allows free replication of our systems, is large enough to train reasonable quality seq2seq models, and has become a de-facto standard database for benchmarking such models. We use chapters 1–49 of the dataset for training, reserving chapter 50 for informal listening and validation during system development.

4. Evaluation

4.1. Listening test

To assess the subjective naturalness of speech synthesised by the systems in the study, we conducted a MUSHRA-like test [20] using audio generated from Harvard sentence prompts [21]. Participants were asked to rate a set of parallel stimuli on a scale from 0 (very poor) to 100 (completely natural). Each set of stimuli were generated from an identical sentence text, but synthesised using all 11 systems listed in in Table 2. Stimuli were presented unlabelled and in random order. As we did not have access to the LJ Speech speaker uttering the Harvard sentences, no reference stimuli were used, and listeners were not required to rate any stimulus at 100. As is common in MUSHRA-like tests for TTS, no explicit lower anchor was used.

24 paid native English listeners, all students at the University of Edinburgh, took part in the evaluation. Listeners were partitioned into 3 groups of 8 listeners each; each group listened to 2 distinct sets of 10 approximately phonetically balanced Harvard sentences, for a total of 480 sets of parallel ratings. All tests were conducted in sound-insulated booths using

Table 3: *Questions for the experiments, relating either to differences between SPSS and sequence-to-sequence neural TTS (primary questions), or differences within paradigms (secondary questions), or how design decisions interact with each other (interaction questions).*

Type	ID	Question: ("What is the impact of…")	Relevant system contrasts
Primary	Q1	…learning the front end?	W2 vs. W2T G1H vs. G1TH G1HA vs. G1THA
	Q2	…acoustic feedback replacing positional feedback?	W2 vs. W2H G1 vs. G1H
	Q3	…jointly-learned alignments replacing fixed alignments?	G1H vs. G1HA G1TH vs. G1THA
Secondary	Q4	…Merlin simplifications to ease stepping toward DCTTS?	M vs. MM
	Q5	…using the DCTTS architecture and codebase?	MM vs. W2
	Q6	…DCTTS waveform generation replacing World?	W2 vs. G2
	Q7	…the DCTTS loss function replacing L2 loss?	G2 vs. G1
Interaction	Q8	Does acoustic feedback interact with the acoustic feature type?	W2→W2H vs. G1→G1H
	Q9	Does front-end learning interact with learning to align?	G1H→G1HA vs. G1TH→G1THA

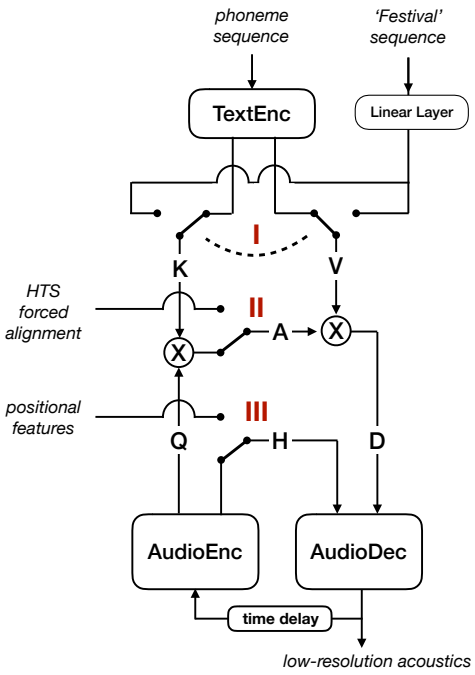

Figure 2: *Block diagram with three switches (in red).*

professional-quality headphones.

For the analysis, each set of 11 ratings was min-max normalised to the range 0 to 100 through an affine transform, to reflect relative differences between the systems. A box plot of the resulting normalised ratings is shown in Fig. 3. Table 4 lists the findings for contrasts between adjacent systems in the study, as graphed in Fig. 1. For each contrast, both listener preferences and the mean difference in normalised rating is tabulated with conf. intervals and p-values adjusted for multiple comparisons.

4.2. Results and discussion

We now review the questions posed in Table 3 in turn and attempt to answer them based on the listening-test results. The very low scores (both absolute and after normalisation) of systems W2, W2T, and W2H mean that differences between these systems should be interpreted with caution. We instead focus more on comparisons featuring other systems, where possible.

Q1: The G1H→G1TH and G1HA→G1THA comparisons show that learning the linguistic analyser helps, and the effects on mean rating and listener preference are substantial. This holds independently of whether attention is used or not.

Q2: The G1→G1H comparison shows that acoustic feedback has a strongly beneficial impact on the quality of synthetic speech. Not all of this improvement can be explained by simplifications made in system G1, since G1H also significantly improves on the original Merlin system M according to Holm-Bonferroni-corrected significance tests like those in Table 4.

Q3: The G1TH→G1THA comparison shows – perhaps surprisingly – that jointly learning to align and predict does *not* significantly improve quality of synthetic speech over using the fixed forced alignment in M, at least for the particular attention mechanism in this study. Furthermore, we can be relatively confident in this conclusion as the hyperparameters we are using are most appropriate at this end of the range of systems, as mentioned at the start of Sec. 3.3.

Q4: The M→MM comparison shows that the simplifications made to the Merlin benchmark in order to compare with systems derived from DCTTS had a disastrous effect on perceived naturalness. Further investigation would be needed to determine whether this is due to the increased frame shift, the decreased number of states in forced alignment, or some interaction of these with other details of model training.

Q5: Stepping from MM to W2 worsened performance further. Again, the number of things that changed in an uncontrolled way at this point in the range of systems means that further investigation would be needed to determine whether this is due to the use different network architectures and sizes, SSRN, different optimisers, learning rates, etc. An omission in the design of systems W2, W2T, and W2H is that no postfiltering or variance expansion was applied to their output (unlike systems M and MM), creating further uncontrolled variation.

Q6: The W2→G2 comparison shows a significant preference for the system employing Griffin-Lim for waveform generation over that employing WORLD. Informal listening shows that the systems have different types of artefact: W2 with its separate F0 representation has a pitch contour which is more properly periodic in voiced speech, but oversmoothed on the whole (which could partly be attributed to the lack of post-processing such as variance expansion). G2's pitch contour is more varied but the speech has a hoarse quality. We attribute this to inappropriate averaging of fine mel-spectral detail erasing most harmonics and effectively lowering the maximum voiced frequency.

Q7: The G2→G1 comparison indicates that no advantage is gained from using L2 loss versus the combined losses employed by DCTTS. However, this comparison takes place between systems on a part of the scale which we treat with caution.

Q8: As expected, results suggest that using acoustic feedback gives greater gains when using mel-spectrogram features (G1→G1H) than when using WORLD features (W2→W2H). The speech produced by system G1 has a hoarse quality due to inappropriate averaging of harmonics as described in Q6 above.

Q9: The fact that the increase in ratings from G1HA→G1THA (where attention is being used) is greater than in the compar-

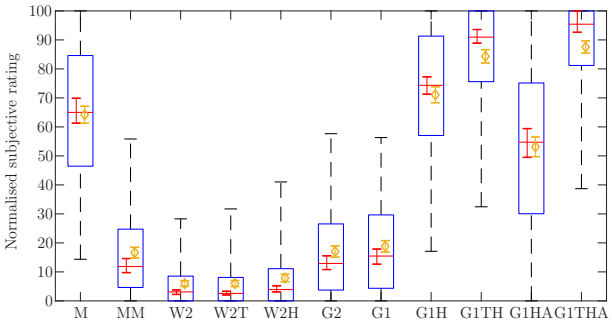

Figure 3: *Box plot of normalised ratings per system. 99% confidence intervals are shown for medians (red lines) and means (yellow diamonds). Whiskers cover 95% of all responses.*

Table 4: *System differences from listening test. 99% confidence intervals and Holm-Bonferroni corrected p-values from pairwise t-tests (mean) and Clopper-Pearson tests (preference ignoring ties). Daggers mark non-significant differences.*

$A \rightarrow B$	$\mathbb{E}[R_B - R_A]$	p-val.	$\mathbb{P}(R_B > R_A)$	p-val.
M→MM	-47.6 ± 3.1	$<10^{-99}$	3%∈(1, 5)	$<10^{-99}$
MM→W2	-10.7 ± 2.0	$<10^{-38}$	21%∈(16,27)	$<10^{-34}$
W2→W2T	0.0 ± 1.4	$0.956^{\dagger}$	49%∈(42,56)	$0.585^{\dagger}$
W2→W2H	1.9 ± 1.6	0.005	56%∈(49,63)	$0.067^{\dagger}$
W2→G2	11.1 ± 2.3	$<10^{-30}$	74%∈(68,80)	$<10^{-25}$
G2→G1	1.8 ± 2.3	$0.075^{\dagger}$	54%∈(47,61)	$0.155^{\dagger}$
G1→G1H	52.3 ± 3.2	$<10^{-99}$	97%∈(94,99)	$<10^{-99}$
G1H→G1TH	13.2 ± 3.4	$<10^{-20}$	72%∈(66,78)	$<10^{-20}$
G1H→G1HA	-17.9 ± 4.0	$<10^{-26}$	29%∈(23,35)	$<10^{-19}$
G1HA→G1THA	34.4 ± 4.1	$<10^{-73}$	87%∈(82,91)	$<10^{-62}$
G1TH→G1THA	3.2 ± 3.3	$0.033^{\dagger}$	56%∈(49,62)	$0.067^{\dagger}$

able case where no attention is used (G1H→G1TH) suggests that naturalness benefits of attention are dependent on a learned front end. We take this to confirm our suspicions that a learned front end is needed to produce an embedding of inputs which is comparable with representations derived from acoustics; the single linear transform of hand-engineered features used by G1HA does not provide the same degree of comparability. Informal listening to the output of system G1HA shows that the speech produced by this system is locally of reasonable quality, but overall marred by skipped and jumbled segments, which we take to indicate failures of the attention mechanism.

5. Conclusions

We proposed (in Table 1) one possible mapping between subnets of recent sequence-to-sequence TTS models (Tacotron 2 and DCTTS) and the components of previous-generation modular SPSS systems (Merlin). We further studied the perceptual implications of the three main characteristics that differentiate the two paradigms. Comparing listener ratings of TTS systems with and without those features, we can conclude that: the use of DCTTS-style attention rather than a fixed alignment does not significantly improve the naturalness of synthetic speech; the use of learned linguistic analysis improves rated naturalness, and seems to be a crucial ingredient in allowing attention to function properly; and finally, conditioning on previously-generated acoustics also leads to significant naturalness gains.

Acknowledgements: Grant support for OW and CVB: EPSRC Standard Research Grant EP/P011586/1; GEH: Swedish Foundation for Strategic Research no. RIT15-0107.

6. References

[1] W. Wang, S. Xu, and B. Xu, "First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention," in *Proc. Interspeech*, 2016, pp. 2243–2247.

[2] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, and Y. Zhang *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. ICASSP*, 2018, pp. 4799–4783.

[3] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," in *Proc. ICASSP*, 2018, pp. 4784–4788.

[4] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From HMMs to DNNs: where do the improvements come from?" in *Proc. ICASSP*, 2016, pp. 5505–5509.

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

[6] A. W. Black, P. Taylor, and R. Caley. (1998) The Festival speech synthesis system. [Online]. Available: http://www.cstr.ed.ac.uk/projects/festival/

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

[8] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[9] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE T. Inf. Syst.*, vol. 99, no. 7, pp. 1877–1884, 2016.

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

[11] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE T. Acoust. Speech*, vol. 32, no. 2, pp. 236–243, 1984.

[12] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, and Y. Xiao *et al.*, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, 2017, pp. 4006–4010.

[13] Y. Wang, D. Stanton, Y. Zhang, R. J. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, and F. Ren *et al.*, "Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis," in *Proc. ICML*, 2018, pp. 5180–5189.

[14] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. EMNLP*, 2015, pp. 1412–1421.

[15] 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.

[16] H. Zen, "An example of context-dependent label format for HMM-based speech synthesis in English," 2006.

[17] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, and D. Povey *et al.*, "The HTK book."

[18] K. Park, "A TensorFlow implementation of DC-TTS," https://github.com/Kyubyong/dc_tts, 2017.

[19] K. Ito, "The LJ Speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[20] *Method for the subjective assessment of intermediate quality levels of coding systems*, ITU Recommendation ITU-R BS.1534-3, International Telecommunication Union, Radiocommunication Sector, Oct. 2015.

[21] E. H. Rothauser *et al.*, "IEEE recommended practice for speech quality measurements," *IEEE T. Acoust. Speech*, vol. 17, no. 3, pp. 225–246, 1969.