Exemplar-based speech waveform generation for text-to-speech

EXAMPLAR-BASED SPEECH WAVEFORM GENERATION FOR TEXT-TO-SPEECH

Cassia Valentini-Botinhao, Oliver Watts, Felipe Espic, Simon King

The Centre for Speech Technology Research, Edinburgh University, UK

ABSTRACT

This paper presents a hybrid text-to-speech framework that uses a waveform generation method based on examplars of natural speech waveform. These examplars are selected at synthesis time given a sequence of acoustic features generated from text by a statistical parametric speech synthesis model. In order to match the expected degradation of these target synthesis features, the database of units is constructed such that the units' target representations are generated from the same parametric model. We evaluate two variants of this framework by modifying the size of the examplar: a small unit variant (where unit boundaries are determined by pitch mark location) and a halfphone variant (where unit boundaries are determined by subphone state forced alignment). We found that for a larger dataset (around four hours of training data) the examplar-based waveform generation variants are rated higher than the vocoder-based system.

Index Terms— Text-to-speech, vocoder, unit selection

1. INTRODUCTION

Hybrid text-to-speech (TTS) systems [1, 2, 3] were, until recent developments in direct waveform prediction [4, 5, 6], the state of the art in natural-sounding speech synthesis and are still widely used in commercial applications. By *hybrid*, we refer to systems that produce speech from waveform unit selection and concatenation, but that guide the selection of units with the output of an acoustic model. In the past, hybrid approaches used hidden Markov model (HMM)-based synthesis as the acoustic model [7, §4], while more recently neural network based models have gained popularity [1, 2, 3]. In most hybrid systems, the units used for waveform generation are relatively large and phonetically determined (diphones and halfphones). To create a database of these units, such systems require a dataset of phonetically transcribed and aligned speech. Alignment is essential for the quality of these systems and when transcription is incomplete, incorrect or simply not available, problems arise. Moreover, the use of larger units makes these systems more dependent on larger datasets, where an appropriate amount of data coverage is obtained. Finally, a waveform generation system that is unaware of the symbolic content underlying the speech signal would in principle be capable of exploiting databases of mixed dialect and languages.

To resolve these issues the current work proposes a hybrid TTS that uses an examplar-based waveform generation method based on smaller units which are determined without phonetic annotation. This waveform generation system was first proposed in [8]; in this paper we integrate it with a TTS acoustic model and present its halfphone variant that was used in [9]. Similar small unit systems have been proposed before, where units are determined without phonetic alignment – these have always been fixed 5 ms frames of speech [10, 11, 12, 13, 14]. Many of these approaches, however, rely on phonetic identity to prune the unit search and reduce computational expense. The work presented in [11, 12] is more similar to ours as no phonetic annotation is assumed. Unlike any other approach, however, we do not use any kind of dynamic programming for unit selection: we find greedy search to be effective when using small units. Furthermore, rather than setting a fixed unit size, we define units according to pitch marks extracted from the speech signal. One example of existing work where unit selection is done over similarly-defined units is [15], although there the units selected (glottal pulses) result in an excitation signal when concatenated, which must then be filtered to impose the vocal tract response heard in the final speech.

In the remainder of this paper we detail the proposed small unit hybrid TTS system and its halfphone variant. Following this, we present results of a listening test with vocoded speech and TTS entries created using two different datasets.

Links to audio samples and code for recreating the systems described here can be found at `https://github.com/CSTR-Edinburgh/snickery`.

2. PROPOSED TEXT-TO-SPEECH SYSTEM WITH EXAMPLAR-BASED SPEECH WAVEFORM GENERATION

The proposed hybrid TTS framework is presented in Fig. 1. At synthesis time a previously trained acoustic model generates a sequence of target acoustic features from text. These features are used to search for a sequence of unit indices. From these indices a sequence of higher dimension acoustic features is created and used for waveform reconstruction. In this section we will summarise the waveform generation method proposed in [8] that forms the basis of the hybrid TTS framework proposed in this paper.

Traditional waveform generation methods are parametric, i.e. they reconstruct the speech waveform from a series of acoustic parameters, relying on models of how the speech signal can be mathematically described. Unlike these more traditional methods, an examplar-based waveform generation method is non-parametric as it recreates the waveform by concatenating waveforms segments derived from a database of natural speech recordings.

As in the case of other examplar-based approaches, the proposed method selects a sequence of speech segments under two types of constraint: that each unit should be acoustically close to its target (divergence is penalised with a *target cost*), and that the end of each unit in the sequence should be acoustically similar to the start of the following unit, so that they can be joined without audible artefacts (implemented with a *join cost*). As mentioned in [8], the target and join components of the combined cost can be regarded as measures of *fidelity* and *fluency* respectively, the first scoring how faithfully the desired message is encoded and the second, how fluently it is rendered. In the following subsections we detail how the database of natural speech waveform units is created and how to generate new waveforms from this database.

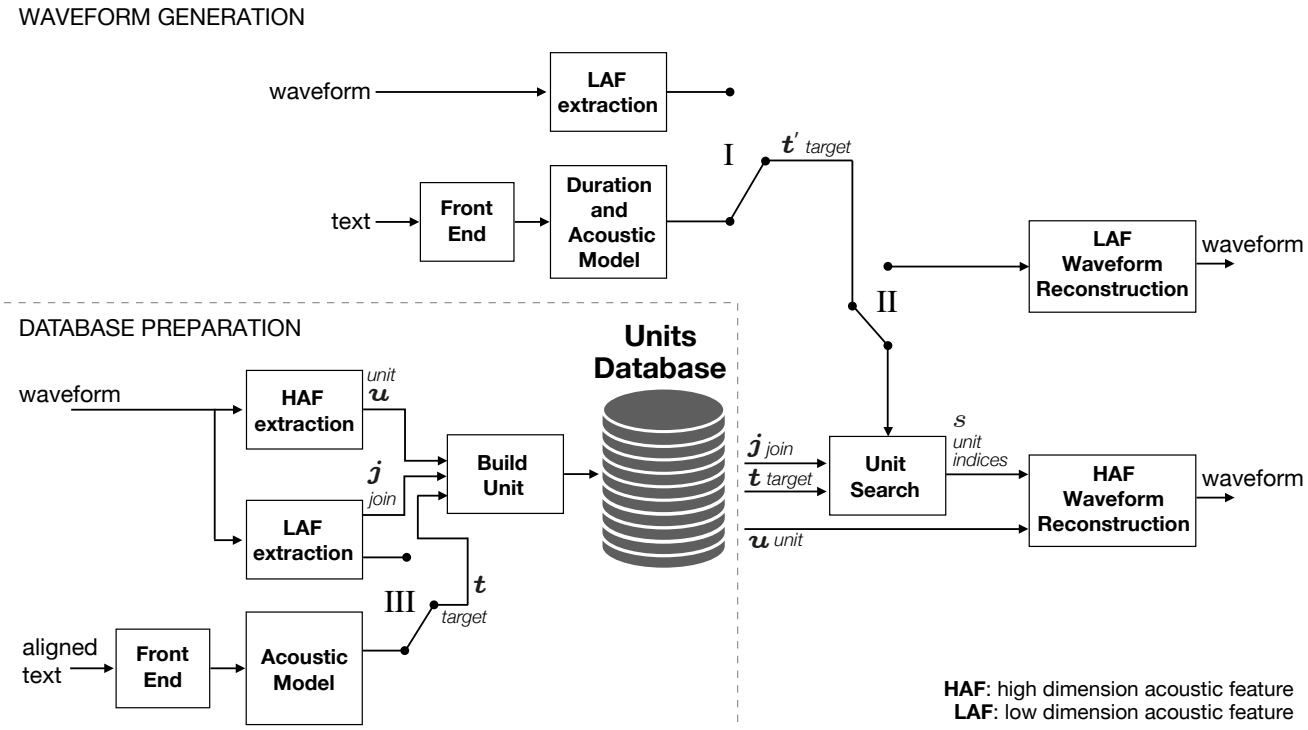

Fig. 1. Proposed hybrid TTS system with examplar-based waveform generation. The switch positions (top/bottom) refer to: switch I (vocoded speech/TTS), switch II (statistical parametric TTS/hybrid TTS) and switch III (matched/unmatched training).

2.1. Database preparation

The process we refer to as database preparation, illustrated in the bottom left corner of Fig. 1, entails two modules: unit building and acoustic feature extraction.

2.1.1. Building units

A unit i is defined by three sets of acoustic representations: the target $\boldsymbol{t}_i$, the join $\boldsymbol{j}_i$ and the unit representation $\boldsymbol{u}_i$. The target and join representations are the basis upon which a fourth representation is created at synthesis time, the combined representation $\boldsymbol{c}_i$. This is the representation used in the unit search module during synthesis. The unit representation is used to reconstruct the waveform from the sequence of unit indices returned by the database search.

2.1.2. Acoustic feature extraction

To obtain these sets of representations we extract a series of acoustic features from a corpus of (possibly transcribed) speech. We first place *pitchmarks* at estimated instants of glottal closure in voiced speech and at 5 ms intervals elsewhere. We then extract spectral features characterising the signal around each of these pitchmarks, through a pitch-synchronous analysis. Following [8], the term *frame* from now on denotes a pitchmark-centred acoustic feature vector.

2.2. Waveform generation

As shown in the bottom right corner of Fig. 1, the waveform generation process is composed of two modules: unit search and waveform reconstruction.

2.2.1. Unit search

Given a database of units, synthesis proceeds as illustrated in Fig. 2. The join and target representations of the units database is used to create the combined representation. This process is illustrated on the top of the Fig. 2, where we see the combined representation $\boldsymbol{c}_i$ for each unit i in the database. This is constructed by concatenating the target representation of unit i $\boldsymbol{t}_i$ with the join representation of the (temporally) preceding unit $\boldsymbol{j}_{i-1}$ as follows:

$$\boldsymbol{c}_i = [\boldsymbol{j}_{i-1}^\intercal \;\; \boldsymbol{t}_i^\intercal]^\intercal \tag{1}$$

At synthesis time step t, the desired combined representation $\boldsymbol{c}'_t$ is prepared by concatenating the desired target vector $\boldsymbol{t}'_t$ and a *history* vector $\boldsymbol{h}$. The desired target feature is obtained from a TTS acoustic model (as shown in Fig. 1). The history vector is the join representation of the unit preceding the previously chosen candidate unit. This process is illustrated in the middle portion of Fig. 2. On the assumption that any sequence to be synthesised will start with silence, the history vector is initialised as the join representation of a frame of acoustic silence.

Given the desired combined representation $\boldsymbol{c}'_t$ and the set of combined representations for each unit in the database, the index s_t of the unit selected at time t is determined as:

$$s_t = \arg\min_i \mathcal{D}(\boldsymbol{c}_i, \boldsymbol{c}'_t) \tag{2}$$

where $\mathcal{D}(\cdot,\cdot)$ denotes Euclidean distance. Once the s_t index is found the history vector $\boldsymbol{h}$ is updated to $\boldsymbol{j}_{s_t}$, and search moves to the next timestep $t+1$. The search is conducted greedly as a fixed decision is made at each time step to find the nearest neighbour in the database.

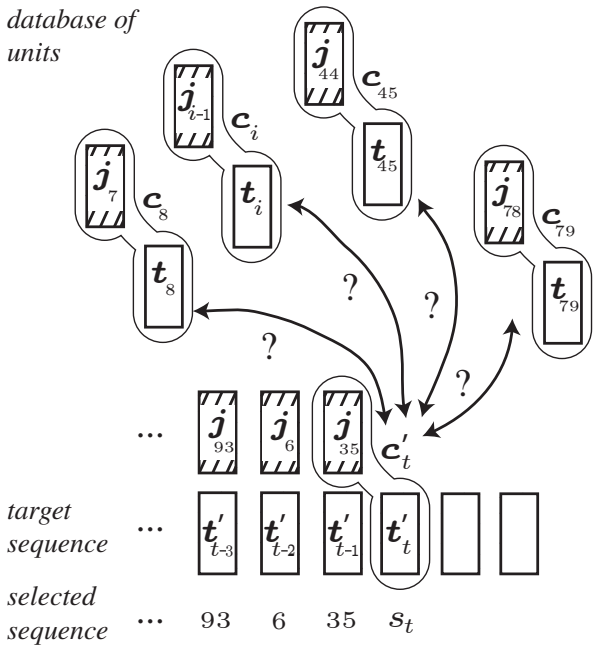

Fig. 2. The unit search module, adapted from [8].

We found that this simple search was sufficient as units are too short to deviate from the target sequence in the course of a single unit [8].

2.2.2. Waveform reconstruction

As illustrated in the Fig. 1, the result of the unit search is a sequence of indices which allow the retrieval of the portion of acoustics associated with each selected unit s_t. In our implementation this is done after search has finished, but as the search requires no lookahead, in principle the concatenation can be done incrementally as search progresses. The streams in the unit representation are cross-faded and waveform is reconstructed using MagPhase synthesis routine based on pitch synchronous overlap-add [16].

2.2.3. Generalisation to multiple frames

The generation modules described in the previous subsections handle units of a single frame (a single epoch). According to our own experiments this creates intelligible speech but results of higher quality can be obtained with slightly larger units. We will now describe the changes made to the unit search module to allow for m-frame unit sizes, where $m > 1$. This process is illustrated in Fig. 3 for m=3. First, the combined representation $\boldsymbol{c}_i$ is obtained as follows:

$$\boldsymbol{c}_i = [\boldsymbol{j}_{i-m}^\intercal \;\; \boldsymbol{t}_{i-m+1}^\intercal \;\; \ldots \;\; \boldsymbol{t}_i^\intercal]^\intercal \tag{3}$$

This is done for every frame in the database. This means that most frames will appear in m different combined representations (i.e. the units overlap temporally).

The desired combined representation $\boldsymbol{c}'_t$ at time t is then prepared by concatenating the history vector $\boldsymbol{h}$ and desired target vectors $\boldsymbol{t}'_{t-m+1} \ldots \boldsymbol{t}'_t$. At each synthesis step, the history vector is updated to $\boldsymbol{j}_{t_s}$ (as it was done for the one frame case). The time index is then incremented from t to $t + m$. This means that contiguous sections of speech consisting of m frames are concatenated in a non-overlapping fashion. The join part of the cost is calculated between single frames of join representation features at unit boundaries.

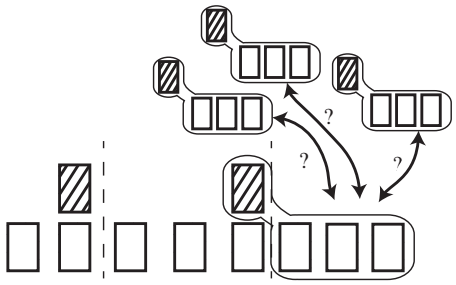

Fig. 3. Generalisation to m-frames (m=3), from [8].

2.3. Acoustic representations

2.3.1. Choosing the target, join and unit representations

We define the target representation as the concatenation of two streams of acoustic features: a one dimensional value corresponding to the logarithm of fundamental frequency ($\log F_0$) and 60-dimensional vector of mel-warped log magnitude spectrum extracted pitch-synchronously by the MagPhase vocoder (mag) [16].

For the join representation four streams of acoustic features were used. In addition to the two streams described above, we also include two streams of phase features extracted by MagPhase (each represented by a 45-dimensional vector). We expect that the inclusion of these two phase streams will yield a sequence of speech fragments with less phase discontinuities.

The unit representation might consist of a fragment of time domain signal or some other perceptually transparent high-fidelity representation of that signal. We defined it as the high dimensional MagPhase representation (in contrast to the low-dimensional features used for search), as this allows for not only high quality reconstruction but also the possibility of applying F_0 manipulation and spectral smoothing at joins. This representation includes: the $\log F_0$, a 1025-dimensional mag feature vector and two vectors of 1025 dimension that encoding the phase information of the complex spectrum.

2.3.2. Standardising the join and target representations

Prior to creating the join, target and combined representation, the separate streams' features are standardised and weighted. To standardise a stream of features we compute means over the whole database per coefficient so that the standardised coefficients will have zero mean. To scale all values within a stream a single standard deviation value is calculated over the whole database and considering all coefficients. This allows for the preservation of the relative dynamic range differences between coefficients within a stream, which we assume are of perceptual importance. When computing means and standard deviation of the $\log F_0$ stream we do not include unvoiced frames. The $\log F_0$ values during unvoiced frames are standardised separately by setting them to a negative constant value. The magnitude of this was set by multiplying the feature's standard deviation by a constant factor.

2.3.3. Weighting the join and target representations

Features' contributions to the selection costs defined in Eq. 2 can be modified by weighting the features that compose the combined representation $\boldsymbol{c}$. We apply weights stream-by-stream rather than coefficient-by-coefficient. The stream weighs in both target and join representations have to sum to one. For the experiments reported here within each representation stream weights were set the to the

same value. We then scale the join representation globally by a factor α, where $0 < \alpha < 1$, and scale the target representation by $1-\alpha$. As in other unit selection approaches, this allows us to strike the right balance between fidelity and fluency.

2.4. The halfphone system

The halfphone variant shares some similarities with the small unit system that we described so far, but the use of larger, phonetically determined and variable length units means that there are considerable differences. Notably, in order to determine unit boundaries, the halfphone systems requires subphone state boundaries to be provided, i.e. phonetic transcriptions are required. Additionally, the halfphone system uses a more conventional Viterbi search to select a sequence of units.

During database preparation, we define halfphones units using a five-state per phone HMM alignment. Two types of halfphone units are possible: left and right halfphones. We assign the speech segment corresponding to the first two states as one halfphone (the left halfphone of this phone), and the speech corresponding to the last three states as another halfphone (the right). The same join acoustic streams used for the small unit system were also used by the halfphone system. The halfphone unit's target representation is extended by appending the duration of the unit. Standardisation and weighting are performed in the same way. In this system, however, frame-level features must be mapped to representations at the rate of the halfphone. To obtain halfphone target representations of fixed size we select three frames from the halfphone frame sequence. The first and last are simply the first and last frames in a given halfphone. The middle frame is not equidistant between those start and end points. Rather, its position is chosen in relation to subphone state boundaries determined during forced alignment, in the expectation that this will provide a more acoustically meaningful point of reference. In practice, we use the last frame of state one as the left halfphone's middle point and the last frame of state four for the right halfphone's midpoint. To obtain the join representation we store frames of join acoustic streams of the start and end of each halfphone. For the unit representation we store references to the start and end samples of the time domain signal. Finally, as well as this numerical data, we store the symbolic phonetic identity of each unit: its quinphone identity and whether it is the left or right halfphone in the phone.

At waveform generation time the input to the unit search module consists of phonetic identities, predicted timings and predicted acoustic features which are used to create acoustic 'targets' for unit selection. Concatenation and normalisation of streams is done as in training, using means and standard deviations computed on the training corpus. The halfphones are then resampled in time to a fixed length, consistent with the representations of units in the training database. Viterbi search of the unit database is carried out. We limit the search space by considering a limited number of candidates γ at each time step. We filtered them according to phonetic type by first taking all units from the database whose quinphone context matches that of the target unit, if any, then do the same for successively more limited contexts: triphone, diphone, and context-independent halfphone, until the desired number of candidates has been selected. In the case of diphone, the direction of context considered depends on whether the target to be matched is the left or right half of a phone. Unit search is treated as a weighted finite-state transducer problem: the target cost is imposed by WFST $\boldsymbol{T}$ and the join cost by $\boldsymbol{J}$. The composition of these produces a WFST whose productions are constrained by both types of cost. The least-penalised path through it is found, corresponding to a sequence of units from the database, whose associated waveform fragments can then be concatenated. At unit concatenation time, the time domain signal corresponding to each halfphone is retrieved and analysed on-the-fly with MagPhase. Following the procedure described in [17], units' spectral representations are then cross-faded in the MagPhase domain, and F_0 trajectories are smoothed before the final speech waveform is synthesised.

Table 1. Experimental conditions.

	Speech	Method	Details and tools
N	natural	-	-
V-MP	vocoded	parametric	MagPhase [16]
V-ES	vocoded	examplar (small unit)	Snickery [8]
T-MP	TTS	parametric	Merlin [21] + MagPhase [16]
T-MS	TTS	unit selection	Multisyn [18]
T-EH	TTS	hybrid (halfphone)	Merlin [21] + Snickery [8]
T-ES	TTS	hybrid (small unit)	Merlin [21] + Snickery [8]

3. EXPERIMENTS

We performed a listening experiment to evaluate the conditions displayed in Table 1. There are three types of speech material in this experiment: natural speech (N), vocoded speech (the 'V-' entries) and TTS (the 'T-' entries). In the following subsections we will detail the databases that were used and how each entry was created.

3.1. Databases

We used two datasets for this experiment. Each dataset contained recordings of read speech from a professional voice talent made in quiet conditions (a semi-anechoid room). The speech material in both datasets have been sampled at 48kHz. Dataset 1 (D1) is made of recordings of a male native speaker of Southern English while the other dataset 2 (D2), contains recordings of a female native speaker of Scottish English. The datasets also differ in terms of their size: D1 contains 83 minutes of audio (1h20min; 2004 sentences) while D2 has 238 minutes (4hrs; 4318 sentences) – around 2.8 times larger.

3.2. Proposed systems and baselines

We evaluate the proposed examplar-based small unit waveform generation system (the '-ES' entries) against three different baselines. One based on the MagPhase waveform generation component, with a vocoder variant (V-MP) and TTS variant (T-MP). The other two baselines are examplar-based TTS systems. A hybrid one, based on halfphone variant described in the previous section (T-EH) and a pure unit-selection TTS system created with the Multisyn tool (T-MS). The Multisyn baseline was constructed following the standard recipe provided by the Multisyn module [18] in the Festival toolkit [19].

We used Reaper [20] to obtain pitch marks and log scale fundamental frequency values (lf0) and MagPhase [16] to retrieve the following pitch synchronous parameters: 60 magnitude (mag) and 90 phase components - 45 of the so-called real and 45 imaginary (imag) features. During unvoiced segments these parameters were extracted at a fixed rate of 5 ms.

3.2.1. Parametric TTS system

To create the TTS models we used the Merlin toolkit [21]. For the D1 data, 1943 sentences were used for training and 61 for valida-

Table 2. Average distortion values of target training features.

Feature	Metric	D1	D2
magnitude	MSE (dB)	7.26	7.68
F_0	MSE (Hz)	9.17	27.20
F_0	Correlation	0.81	0.83
V/UV	Decision error (%)	0.03	0.04

Table 3. Statistics of units selected for the test set.

	D1	D2
voiced unit size (ms)	47.71	58.18
unvoiced unit size (ms)	3.00	6.00
number of joints per second	21.30	15.60

tion, while for D2, 4217 and 101 sentences were used respectively. The models of both voices were trained using the same architecture: six layers of 1024 'tanh' units each. For training, we used 0.002 learning rate, a batch size of 256, 25 epochs and stochastic gradient descent as the optimiser. The learning rate and optimiser were varied in order to find the best parameter for each dataset. We used as output features the following acoustic streams: mag, real, imag, interpolated lf0 and vuv (voicing decision derived from lf0). As input we used 601 linguistic features derived from text plus 9 position and duration features. The duration features were extracted from state aligned labels during training and predicted from text at synthesis time using a duration model. The duration model trained for each dataset had the same architecture and training configuration as the acoustic model, apart from the batch size of 64. The same 601 linguistic features were used as input and as output 5 duration features were used following the standard recipe in [21].

Table 2 shows the average distortion of the target feature streams generated by the SPSS model. This was calculated per frame over the training and validation sentences. The reference used were acoustic features extracted from natural speech. We can see that the models are relatively comparable in terms of magnitude spectrum errors but that fundamental frequency mean square errors were much higher for system D2.

3.2.2. Proposed hybrid TTS systems

To create the target representation of the database of units for the proposed hybrid systems we synthesised the training and validation sentences using natural speech duration and the linguistic features as the input to the trained TTS acoustic model. The generated acoustic features created from these models were used to compose the target features that describe each unit so that at training and synthesis time the conditions are matched. The join and unit representations used were extracted from natural speech, as shown in Fig.1. The database preparation process resulted in a database with approximately 760,000 units for D1 and 2,910,000 units for D2.

To fine tune the proposed hybrid systems we synthesized 19 new sentences (taken from the phonetically balanced Harvard set [22]) using a selection of different settings. Conditions were compared pairwise and the best condition was chosen for the evaluation. This process did not take more than ten minutes per voice. The different settings were created by varying two parameters. For the halfphone voices, we varied the join cost weight α (from 0.5 to 0.9, with steps of 0.1) and the number of candidates for pre-selection γ (from 30 to 70, with steps of 10). For the small unit voices, both α (from 0.1 to 0.5, step of 0.1) and unit size m (from 4 to 12 epochs, steps of 2) were varied. The final selected parameters for the small unit systems

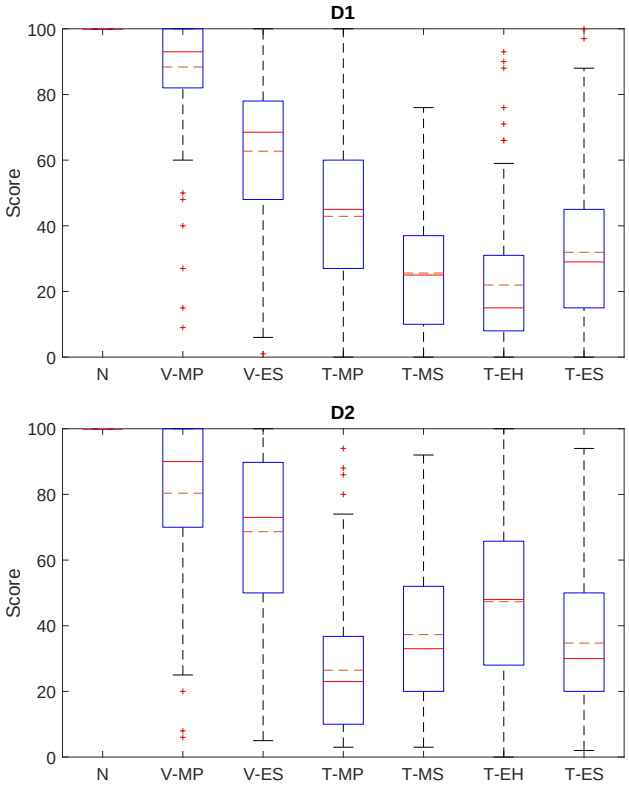

Fig. 4. Boxplot of listening test scores for conditions created using database D1 (top) and database D2 (bottom).

were: m=6 and α=0.2 for D1; and m=12 and α=0.2 for D2. For the halfphone systems the D1 voice was constructed with γ=50 and α=0.7, while the D2 system, γ=70 and α=0.7.

Table 3 shows the average unit size observed in the test set. As units are defined by a multiple of the pitch period, the size of voiced units is inversely proportional to the underlying F_0 of that segment. For this reason the average unit size for D2 (female speaker) is only 1.2 times bigger than D1 even though the m value chosen for that voice was twice as big. For unvoiced segments, the unit are defined as m times 5ms, i.e. unvoiced units in D2 are twice as long as D1 units. This explains why the number of joins made per second (excluding natural joins and silence) was higher for the D1 entry.

3.3. Listening experiment design

We created a MUSHRA-style test [23] with 22 screens. On each screen participants could play the audio sample produced by different systems for the same sentence as many times they wished. They were asked to rate the quality of the samples from 0 (bad) to 100 (excellent). The screens in the first half of the experiment were made of samples from D1, while the second half contained samples from D2. The first screen in each half was used for training. A different sentence was used for each screen such that across every four listeners, 40 different sentences were used for each voice. Natural speech (N) was included on each screen so that participants would have a quality reference and to check if participants paid sufficient attention to score it as 100 (as instructed). We recruited 22 native English speakers. One participant was excluded as they rated N less than 100% in at least 20% of screens for both voices. We excluded around 11% of screens where listeners did not give N the highest score.

3.4. Results

We present the boxplot of the results in Fig. 4. Median and mean values of each distribution of scores are presented with solid and dashed lines. As a significance test we used a Mann-Whitney U test, at a p-value of 0.05, and with Holm Bonferroni correction. All systems were perceived to be significantly different from each other except T-MS and T-ES for the D2.

As illustrated in Fig. 4, among the vocoded entries V-MP obtained the highest mean scores for both datasets, followed by the proposed small unit system (V-ES). These results are in agreement with those obtained in [8].

For the TTS entries results vary considerably depending on the dataset. For D1 (the male speaker smaller dataset), the vocoder-based voice T-MP was rated highest followed by T-ES, T-MS and T-EH. For the D2 (the larger female speaker dataset), results are the opposite: the higher score is obtained by T-EH, followed by T-MS, T-ES and T-MP.

3.5. Discussions

The final quality of a voice produced by any hybrid TTS system will depend on the quality of both the acoustic model and the unit-selection module. The results obtained by T-MP gives us an indication of the quality of the underlying acoustic model for each dataset. If we assume we can compare results across datasets (tentatively, as no anchors were used in the test), we find that the T-MP system is rated much lower when trained with D2 – even though more training data was available, a result that was also found in [16] using the same datasets and partially supported by results in Table 2. The improvements observed for this dataset when using the hybrid systems (T-ES and T-EH) could reflect two things: the benefit of having a larger amount of units to choose from (the D2 database is 3.8 times bigger than D1) and the benefit of using a waveform generation module that can compensate for imperfect acoustic targets.

In terms of understanding what type of unit results in higher quality, what we observed is that even though the performance of the halfphone system (T-EH) was better in the D2 case, it varied greatly by dataset. This could indicate that this system is more dependent on data coverage, as we expect of a system built with larger units. More experiments are required to determine whether more (possibly untranscribed) data can improve the performance of T-ES, particularly for D1, and to determine whether the benefits observed for D2 were mainly the result of the greater amount of available units.

4. CONCLUSIONS

We presented a hybrid text-to-speech system based on an unit-selection waveform generation method. This method utilises units defined by acoustics only, not relying on phonetic transcription or alignment. In order to preserve perceptually relevant segments, the unit size is defined by multiples of pitch marks. To guide unit selection at synthesis time we use the output of a neural network based TTS acoustic model. We conducted an evaluation comparing this system with a halfphone variant and with a system where waveforms are reconstructed with a deterministic vocoder. Results varied according to the dataset. For a larger dataset of a female speaker's speech the halfphone variant was preferred, while for the other dataset it was the vocoder-based system.

Acknowledgements: This research was supported by EPSRC Standard Research Grant EP/P011586/1 *Speech Synthesis for Spoken Content Production (SCRIPT)*.

5. REFERENCES

[1] Thomas Merritt, Robert A.J. Clark, Zhizheng Wu, Junichi Yamagishi, and Simon King, "Deep neural network-guided unit selection synthesis," in *Proc. ICASSP*, 2016, pp. 5145–5149.

[2] Tim Capes, Paul Coles, Alistair Conkie, Ladan Golipour, Abie Hadjitarkhani, Qiong Hu, Nancy Huddleston, Melvyn Hunt, Jiangchuan Li, Matthias Neeracher, Kishore Prahallad, Tuomo Raitio, Ramya Rasipuram, Greg Townsend, Becci Williamson, David Winarsky, Zhizheng Wu, and Hepeng Zhang, "Siri on-device deep learning-guided unit selection text-to-speech system," in *Proc. Interspeech*, August 2017.

[3] Vincent Wan, Yannis Agiomyrgiannakis, Hanna Silen, and Jakub Vit, "Google's next-generation real-time unit-selection synthesizer using sequence-to-sequence LSTM-based autoencoders," in *Proc. Interspeech*, August 2017.

[4] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," .

[5] Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio, "SampleRNN: An Unconditional End-to-End Neural Audio Generation Model," .

[6] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu, "Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions," .

[7] Heiga Zen, Keiichi Tokuda, and Alan W. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[8] Oliver Watts, Cassia Valentini-Botinhao, Felipe Espic, and Simon King, "Exemplar-based speech waveform generation," in *Proc. Interspeech*, September 2018.

[9] Felipe Espic, Avashna Govender, Sam Ribeiro, Cassia Valentini-Botinhao, and Oliver Watts, "The CSTR entry to the 2018 Blizzard Challenge," in *Proc. Blizzard Challenge Workshop*, India, September 2018.

[10] Zhen-Hua Ling and Ren-Hua Wang, "HMM-based unit selection using frame sized speech segments," in *Proc. Interspeech*, 2006.

[11] Toshio Hirai, Junichi Yamagishi, and Seiichi Tenpaku, "Utilization of an HMM-based feature generation module in 5 ms segment concatenative speech synthesis," in *Proc. SSW*, August 2007.

[12] Yao Qian, Ji Xu, and Frank K. Soong, "A frame mapping based HMM approach to cross-lingual voice transformation," in *Proc. ICASSP*, May 2011, pp. 5120–5123.

[13] Yao Qian, Frank K. Soong, and Zhi-Jie Yan, "A unified trajectory tiling approach to high quality speech rendering," *IEEE Trans. on Audio, Speech and Language Processing.*, vol. 21, no. 2, pp. 280–290, 2013.

[14] Zhi-Ping Zhou and Zhen-Hua Ling, "DNN-based unit selection using frame-sized speech segments," in *Proc. Int. Symp. on Chinese Spoken Lang. Proc.*, Oct 2016, pp. 1–5.

[15] Tuomo Raitio, Antti Suni, Junichi Yamagishi, Hannu Pulakka, Jani Nurminen, Martti Vainio, and Paavo Alku, "HMM-based speech synthesis utilizing glottal inverse filtering," *IEEE Trans. on Audio, Speech and Language Processing.*, vol. 19, no. 1, pp. 153–165, Jan 2011.

[16] Felipe Espic, Cassia Valentini-Botinhao, and Simon King, "Direct modelling of magnitude and phase spectra for statistical parametric speech synthesis," in *Proc. Interspeech*, August 2017.

[17] Srikanth Ronanki, Sam Ribeiro, Felipe Espic, and Oliver Watts, "The CSTR entry to the Blizzard Challenge 2017," in *Proc. Blizzard Challenge Workshop*, Stockholm, Sweden, August 2017.

[18] Robert A.J. Clark, Korin Richmond, and Simon King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317 – 330, 2007.

[19] Paul Taylor, Alan W Black, and Richard Caley, "The architecture of the festival speech synthesis system," in *Proc. ESCA workshop in speech synthesis*, 1998, pp. 147–151.

[20] "REAPER: Robust Epoch And Pitch EstimatoR," `https://github.com/google/REAPER`, 2017.

[21] Zhizheng Wu, Oliver Watts, and Simon King, "Merlin: An open source neural network speech synthesis system," in *Proc. SSW*, Sept. 2016, pp. 218–223.

[22] IEEE, "IEEE recommended practice for speech quality measurement," *IEEE Trans. on Audio and Electroacoustics*, vol. 17, no. 3, pp. 225 – 246, 1969.

[23] International Telecommunication Union Radiocommunication Assembly, Geneva, Switzerland, *Method for the subjective assessment of intermediate quality level of coding systems*, March 2003.