Text-to-speech duration models for resource-scarce languages in neural architectures

Johannes A Louw[0000−0002−8168−7857]

Voice Computing Research Group,
Next Generation Enterprises and Institutions, CSIR,
Pretoria, South Africa
jalouw@csir.co.za

Abstract. Sequence-to-sequence end-to-end models for text-to-speech have shown significant gains in naturalness of the produced synthetic speech. These models have an encoder-decoder architecture, without an explicit duration model, but rather a learned attention-based alignment mechanism, simplifying the training procedure as well as the reducing the language expertise requirements for building synthetic voices. However there are some drawbacks, attention-based alignment systems such as used in the Tacotron, Tacotron 2, Char2Wav and DC-TTS end-to-end architectures typically suffer from low training efficiency as well as model instability, with several approaches attempted to address these problems. Recent neural acoustic models have moved away from using an attention-based mechanisms to align the linguistic and acoustic encoding and decoding, and have rather reverted to using an explicit duration model for the alignment. In this work we develop an efficient neural network based duration model and compare it to the traditional Gaussian mixture model based architectures as used in hidden Markov model (HMM)-based speech synthesis. We show through objective results that our proposed model is better suited to resource-scarce language settings than the traditional HMM-based models.

Keywords: HMM · DNN · Speech synthesis · duration modelling · resource-scarce languages

1 Introduction

Deep neural network (DNN) based techniques applied to text-to-speech (TTS) systems have brought on dramatic improvements in the naturalness and intelligibility of synthesized speech. An example of the change in the landscape could be seen in the 2019 edition of the Blizzard Challenge [1], where the best perceptually judged entry was based on a long short-term memory (LSTM) - recurrent neural network (RNN) hybrid architecture [4] with WaveNet [22] as the vocoder. In fact, of the twenty one entries to the Blizzard Challenge 2019 that submitted an accompanying paper (on the Blizzard Challenge website[1]), one system

[1] http://festvox.org/blizzard/blizzard2019.html

was based on a traditional unit-selection architecture, one system was based on a hidden Markov model (HMM) - deep neural network (DNN) hybrid driven unit-selection architecture, one system was based on a HMM-DNN based hybrid architecture, whilst the other eighteen systems were based on some or other DNN architecture.

The current research in TTS is dominated by DNN-based architectures as can also be seen from the paper submissions to the 2019 edition of Speech Synthesis Workshop (SSW)[2]. According to [24], the success of these architectures in the improvement of the synthesized speech naturalness and intelligibility can be broadly attributed to the attention-based models (such as Tacotron [23] and Deep Convolutional TTS (DCTTS) [19]) as well as the use of neural network based vocoders (such as WaveNet [22])).

Many of the newer DNN-based sequence-to-sequence model architectures are what is known as "end-to-end" systems, in that they only require text and audio pairs (<text, audio>) for training. The traditional TTS architectures are usually based on a pipeline of a linguistic front-end and a waveform generation back-end, requiring specialized linguistic knowledge or engineering capabilities for building new voices.

A major challenge of the end-to-end architectures is the computational complexity and load, where for example the WaveNet vocoder achieves a 0.3× real time synthesis speed of 16-bit 24kHz mono audio on a Nvidia P100 GPU, and WaveRNN [5], which aims to improve the synthesis speed, achieves a 4× real time speed of the same fidelity audio on the same hardware [5]. This improvement still represents a very high computational load. The training time and computational requirements is also something that needs to be taken into consideration, for example the Tacotron 2 architecture [16] takes on average 234 hours[3] to train (at 32-bit floating point precision) whilst the WaveGlow vocoder [16] (WaveGlow is one of the newer vocoders that was developed to address the high computational requirements of WaveNet) takes on average 768 hours[3] to train (also at 32-bit floating point precision). These performance numbers were obtained with one Nvidia V100 16G GPU, which at the time of writing costs in the region of $6000 - $7000 each (excluding supporting hardware, importing costs and taxes).

Other challenges of the end-to-end models are that the attention-based alignment systems, such as used in the Tacotron, Tacotron-2, Char2Wav [18] and DC-TTS [19] architectures, typically suffer from low training efficiency as well as model instability [30]. The low training efficiency means that one requires more data than is usually available for low-resourced environments (DC-TTS in [19] used 24 hours of data, which still resulted in reverbed quality synthesized speech). Model instability may happen due to inaccurate alignments by the attention mechanism, resulting in repeated, skipped or mispronounced phonemes or words.

[2] https://www.isca-speech.org/archive/SSW_2019/

[3] https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/\SpeechSynthesis/Tacotron2#expected-training-time

Recent neural acoustic models such as Fastspeech [14], FastSpeech 2 [13] and a bottleneck feed-forward neural network implemented in [8] have moved away from using attention mechanism to align the linguistic and acoustic encoding and decoding, and have rather reverted to using an explicit duration model for the alignment. With our focus being on developing and implementing DNN architectures for resource-scarce environments we are looking at duration models in this work, and in particular speaker specific or dependent models. We compare the traditional HMM-based duration models with a DNN-based model suitable for resource-scarce environments and report on objective measures between the two models and a reference data set.

The organisation of the paper is as follows: in Section 2 we give some background on duration modeling as well as an overview of the two approaches followed in this work. Section 3 details our experiments and results, and lastly a discussion and conclusion is presented in Section 4.

2 Duration Models

Duration models, or to be more precise, phonetic duration models are employed in a TTS pipeline architecture in order to inform the phonetic acoustic model of the number of acoustic frames for which it must generate acoustic parameters, or features, that will typically be synthesized by a downstream vocoder into synthetic speech.

Intonation, emphasis or prominence and phrasing are influenced by the duration of the different phonetic units of an utterance [20]. The dynamic properties of the phonetic unit durations and their relationships and interactions in an utterance are complex, for example if one talks faster then the factor of speed increase is not applied equally to all phonetic units.

Early formant and diphone based TTS systems used sets of deterministic rules [7] developed by linguistic experts. Some models used the syllable as the fundamental unit of duration [2], as syllables are believed to be the natural units of prosody [20]. Data driven techniques for phonetic duration modelling have become ubiquitous, including decision trees [15], neural networks [25] and genetic algorithms [11].

In this work we will be comparing an HMM- and DNN-based duration model in a resource-scarce setting.

2.1 HMM-based Duration Models

The technique we describe here is based on the widely used *HMM-based Speech Synthesis System* (HTS) [28]. The fundamental unit of duration is a phoneme, and each phoneme is modelled as a 5-state left-to-right, with no skip, HMM. State duration densities are modeled by single Gaussian distributions.

The duration models are context dependent, with many contextual factors that influence the duration of the individual phonemes taken into account (e.g.,

phone and phone context identity factors, stress-related factors, locational factors). The contextual factors taken into account depend on their availability in the particular language in question, i.e. some resource-scarce languages for example might not have any available stress models.

During training, a decision-tree based context clustering technique is used to cluster states of the context dependent HMMs. The decision-tree has a question at each node which splits the context into two groups (i.e. a binary tree). The clustered context dependent states are tied (shared) and are reestimated with embedded training.

During synthesis, the target text to be synthesized is converted to a context-based label sequence by the TTS engine front-end. A sentence HMM is constructed by concatenating context dependent HMMs according to the label sequence. The state durations of the sentence HMM can then be determined from the total length of speech and the state duration densities.

The reason for the decision-tree based context clustering technique is to overcome data scarcity, as it is impossible to prepare a speech database which includes all combinations of contextual factors.

Figure 1 shows the synthesis steps, where the decision- or regression tree is used to select the context dependent HMMs based on the context labels of the target text. The HMMs are concatenated to form the HMM sentence, which can then used to determine the phoneme durations from the HMM state durations.

2.2 DNN-based Duration Models

The DNN-based phonetic duration model used in this work is based on a stack of fully connected layers in a feed-forward neural network (FFNN), as given in figure 2. At the output is a linear layer, whilst the *rectified linear unit* (ReLU) activation function was used for the hidden layers. Batch normalization and dropout were used with each hidden layer of the network. The Adam optimisation algorithm [6] was used with a learning rate scheduler that lowers the learning rate when the validation loss reaches a plateau (the Adam optimisation algorithm adjusts the learning rate, it is the upper bound that we reduced). The weights and biases of all the layers were initialized using the *He-uniform distribution* [3]. The loss function was the mean squared error (MSE) on the predicted duration feature.

As with the HMM-based duration models of the previous section, the features used included many contextual factors that influence the duration of the individual phonemes.

During training the TTS engine front-end creates a contextual label sequence for each recording of the training data in the speech database. This contextual label sequence is then converted into a linguistic description feature vector, used as input to the FFNN. The ground truth duration of each phone unit in the contextual label sequence is taken from the recorded database and used as the output feature target of the FFNN.

During synthesis the TTS engine front-end creates a contextual label sequence for the target utterance. This contextual label sequence is then converted

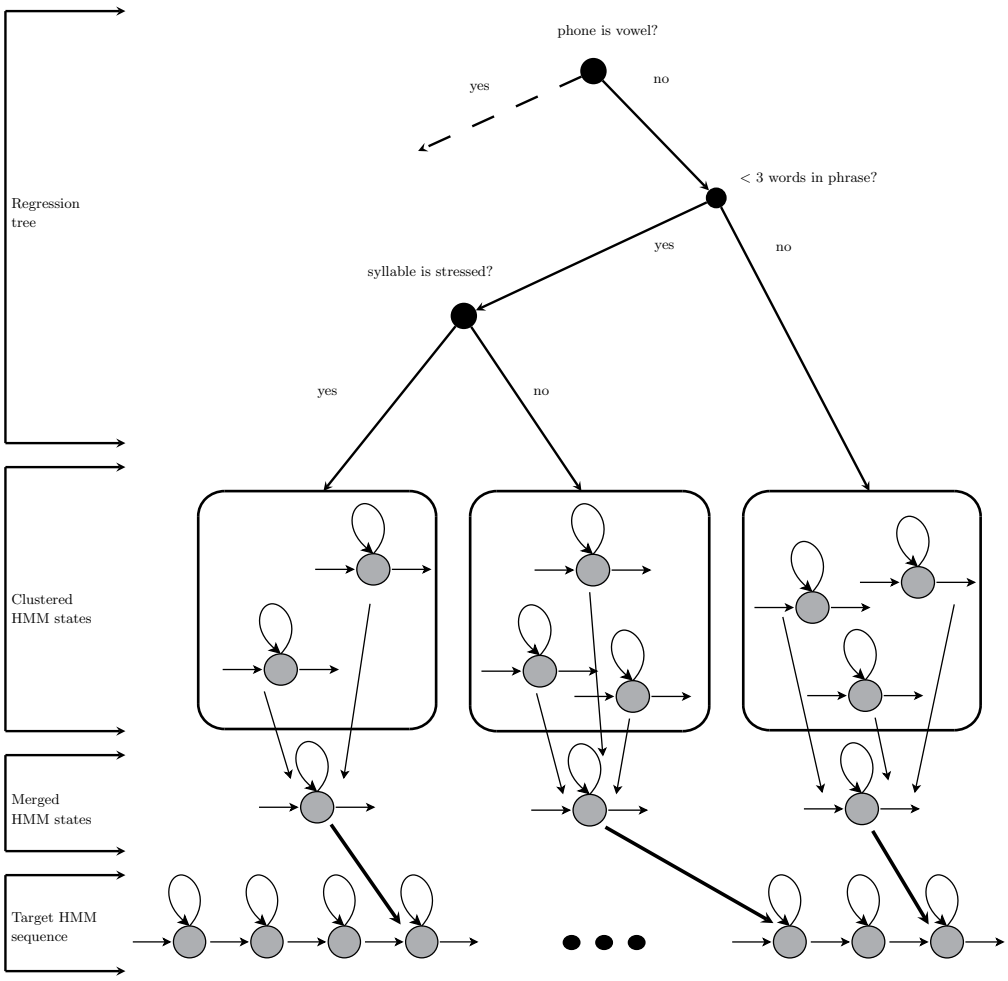

Fig. 1. HMM-based duration model.

into a linguistic description feature vector, used as input to the FFNN, and the FFNN does a prediction of the duration at the output.

3 Experimental Setup

3.1 Data

The data used in this work is a *subset* of an in-house single speaker Afrikaans female TTS corpus of duration 12:08:15.89. The corpus was recorded in a studio with a professional voice artist at a 44.1 kHz sampling rate with 16 bits precision. The subset used are recordings of the text of the *Lwazi II Afrikaans TTS Corpus* [12], consisting of 763 utterances of duration 00:56:30.29. This subset

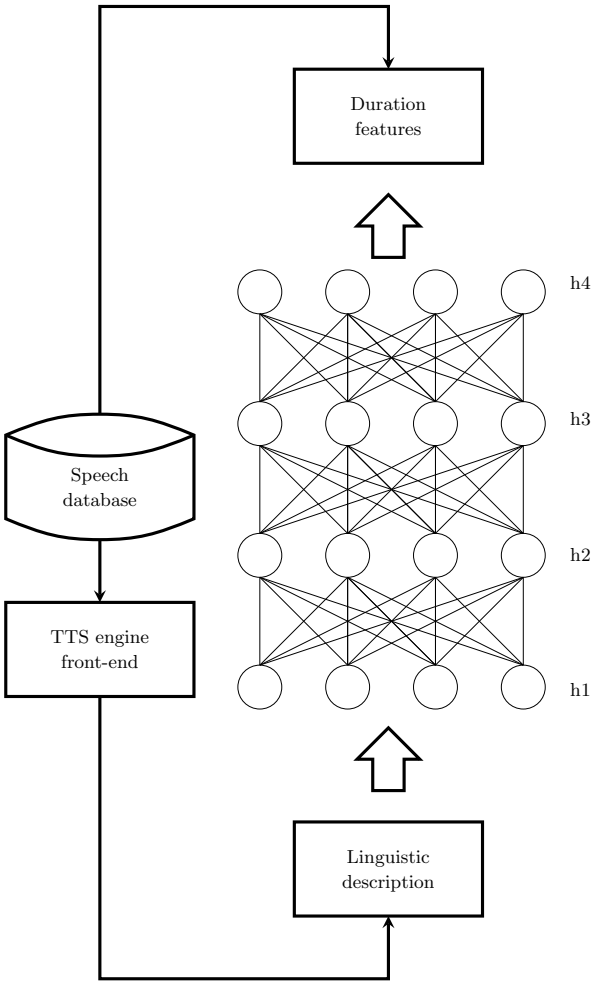

Fig. 2. DNN-based duration model.

represents a small and phonetically balanced speech database as would be used for building HMM-based synthetic voices and attempting to build DNN-based synthetic voices.

The utterances were randomly split into training, validation and testing sets as given in Table 1. All audio was down-sampled to 16 kHz at 16 bits per sample and each utterance was normalised to the average power level of the subset (the 763 utterances).

Contextual Features The text annotations of the speech database (Table 1) were tokenized and normalised with the Speect TTS engine front-end [10]. The context features used in this work is given in Table 2, and these features are the

Table 1. Speech database utterance splits as used in the experiments.

Set	# Utterances	Duration
Training	715	00:53:12.09
Validation	38	00:02:37.99
Test	10	00:00:40.21

same as defined in [21], except for syllable stress, accent and ToBI (*Tones and Break Indices*) [17] tones which were not included due to it most probably not being available in resource-scarce settings. The context features of each utterance was also extracted using Speect.

Table 2. The linguistic context features as used in this work.

Context	Feature
Phoneme	the current phone
	the two preceding and succeeding phones
	the position of the current phone within the current syllable
Syllable	the number of phonemes within preceding, current, and succeeding syllables
	the position of the current syllable within the current word and phrase
	the number of preceding and succeeding stressed syllables within the current phrase
	the number of preceding and succeeding accented syllables within the current phrase
	the vowel identity within the current syllable
Word	guessed part-of-speech (GPOS) of preceding, current, and succeeding words
	the number of syllables within preceding, current, and succeeding words
	the position of the current word within the current phrase
	the number of preceding and succeeding content words within the current phrase
	the number of words from the previous content word
	the number of words to the next content word
Phrase	the number of syllables within preceding, current, and succeeding phrases
	the position of the current phrase in major phrases
Utterance	the number of syllables, words, and phrases in the utterance

Reference Durations The reference durations of the phone units in the speech database (Table 1) were obtained from a forced-alignment procedure using the HTK toolkit [27]. A frame resolution of 10ms was used (hop size). A *silence*

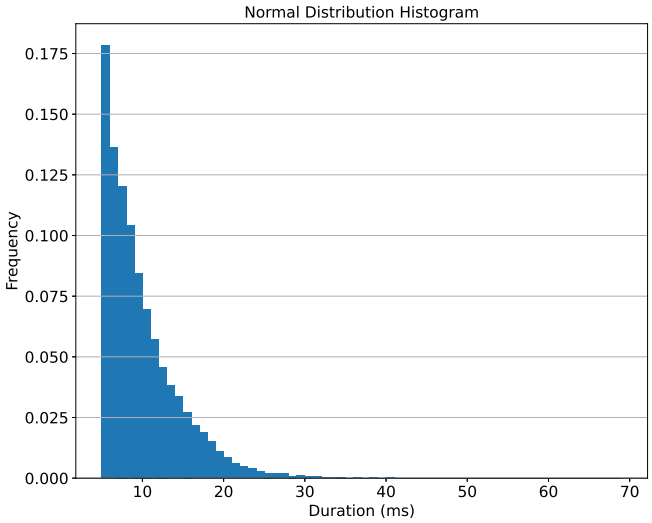

Fig. 3. The phone duration distribution of all the non-silent phones in the speech database.

state was added between all words in the database in order to identify any pauses or phrase breaks which were recorded but not specifically annotated in the text with punctuation marks (based on work in [9]). Any non-annotated silence longer than 80ms is marked as a pause and a phrase break is inserted into the utterance structure at this point. These phrase breaks have an influence on the context features as given in Table 2.

Figures 3 and 4 give the duration distributions of all the non-silent phones and the near-open front unrounded vowel (/æ/) respectively. Note that the minimum phone duration is 5 frames due to the use of a 5-state HMM model (see Section 2.1).

3.2 HMM-based Duration Model

After the reference durations were extracted, a duration model was built based on the standard architecture of 5-state (excluding the emitting states), left-to-right HMM. The contextual features used were as defined in Table 2. The duration features were modelled by a single-component Gaussian. The decision trees state clustering was done using a minimum description length (MDL) factor of 1.0. Training of the model was done via custom scripts based on the standard demonstration script 2 available as part of HTS [33] (version 2.2).

Note that the model was only trained on the 715 training utterances of Table 1.

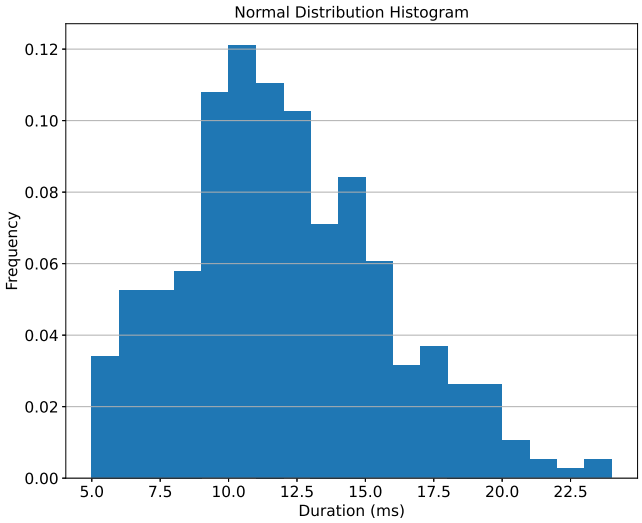

Fig. 4. The phone duration distribution of all the near-open front unrounded vowel (/æ/) in the speech database.

3.3 DNN-based Duration Model

The contextual features of Table 2 were converted to a linguistic description vector containing a combination of binary encodings (for the phoneme identities and features) and positional information (as is done in [26]).

The input linguistic descriptions vector consisted of 375 features and was normalised to the range of [0.01, 0.99], whilst the output vectors (the reference durations) were normalised to zero mean and unit variance. TTS is a highly unbalanced mapping problem when viewed as a sequence-to-sequence mapping model [30] (mapping text to speech frames). The output speech sequence is much longer than the input text sequence. In order to add granularity on the text side, the durations are modeled in terms of their "HMM" states, i.e. as if the model consists of a number of HMM states. This has been proven to improve the quality of the synthesized speech [29]. The output vector has a normalised frame duration for each "state" of the HMM model (which was modelled using a 5-state left-to-right HMM).

Note that this model was also only trained on the 715 training utterances of Table 1, the same as the HMM model. Various model hyper-parameters in terms of the number of hidden layers and the number of units per hidden layer were trained and all the results are given in Table 3.

3.4 Results

The validation and test sets of Table 1 were synthesized with the HMM- and DNN-based duration models and the durations per phone unit were extracted.

Objective Measures Two objective measures were used in order to evaluate how closely the models were able to predict the durations of the phonemes of the particular speaker. The Pearson correlation coefficient and the Root Mean Square Error (RMSE) between the predicted (y) and actual (x) durations (in terms of number of frames) were calculated.

The Pearson correlation coefficient (ρ) is given by:

$$\rho_{xy} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \tag{1}$$

and the RMSE by:

$$RMSE = \sqrt{\frac{\sum_i (x - y)^2}{T}} \tag{2}$$

where T is the number of frames. The results of the objective measurements on the synthesized durations of the validation and test sets of various architectures are given in Table 3. Higher correlation (ρ) is better whilst lower $RMSE$ (in terms of frames/phone) is better.

Table 3. Results of objective measurements for different model architectures. Root mean squared error ($RMSE$) is in units of frames per phone whilst the Pearson correlation coefficient (ρ) is dimensionless.

Architecture	**Validation set**		**Test set**	
	$RMSE$	ρ	$RMSE$	ρ
HMM, 5-state, single Gaussian	4.288	0.633	6.644	0.552
6 hidden layers, 128 units/layer	3.819	0.696	3.067	0.827
6 hidden layers, 256 units/layer	3.797	0.702	4.685	0.707
6 hidden layers, [512, 256, 128, 64, 32, 16] units	3.801	0.709	3.174	0.809
4 hidden layers, 128 units/layer	3.773	0.702	**2.905**	**0.832**
4 hidden layers, 256 units/layer	3.771	0.709	3.052	0.812
4 hidden layers, [128, 64, 32, 16] units	**3.739**	**0.720**	3.240	0.778

Figure 5 shows a visual comparison between the durations on a word level predicted by a HMM model, a DNN model and the reference recording.

4 Discussion and Conclusion

In this work we have developed an efficient feed-forward neural network for speaker dependent phonetic duration modeling in the context of resource-scarce

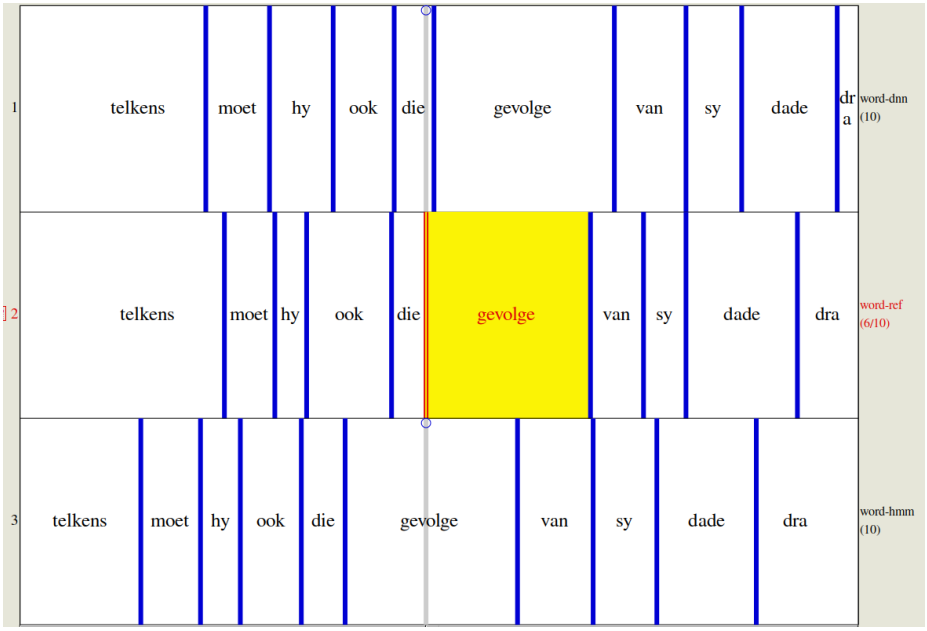

Fig. 5. A visual comparison of the duration prediction on the word level for the utterance *"Telkens moet hy die gevolge van sy dade dra"*. At the top is the DNN prediction, at the bottom the HMM prediction and in the middle the reference from the recorded speech.

text-to-speech settings. Our model trains in less than 2 hours on a CPU and therefore can be easily adapted. Although not reported on, we have also applied this model on isiXhosa, isiZulu as well as Setswana corpora with similar success.

The importance of these types of explicit duration models have declined with the advent of the attention-based mechanisms in end-to-end neural speech synthesis architectures such as Tacotron, Tacotron 2 and Char2Wav. However, the challenges brought on with the attention-based mechanisms architectures and their unsuitability in resource-scarce environments have prompted the development of acoustic models such as Fastspeech and Fastspeech 2, which again use explicit duration models.

Our results show that a simple FFNN, with 4 hidden layers, can accurately predict phone unit duration and can reach a RMSE of 2.905 frames/phone on a speech database of less than 1 hour in duration, with a high correlation over the whole sequence of phones.

It is interesting to note that larger networks do not necessarily perform better, which may be attributed to the lack of data for training the larger systems.

In contrast to our work, [20] mentioned that previous comparative studies between decision trees and neural networks found little difference in accuracy between either approach. We think that this may be due to the size of the dataset

used for training, as it has been shown that neural networks with the appropriate architectures are much more data efficient than HMMs [29] and the dataset used in this work is particularly small.

Future work will include variants of output layers, such as a softmax function, to predict a region wherein the duration of a specific input may lie, such as done in [25]. The importance of specific linguistic features are also of particular interest, as eliminating hand crafted or expert developed features make it easier to develop voices in new languages.

References

1. Black, A.W., Tokuda, K.: The Blizzard Challenge-2005: Evaluating corpus-based speech synthesis on common datasets. In: Ninth European Conference on Speech Communication and Technology. pp. 77–80 (September 2005)
2. Campbell, W.N.: Syllable-based segmental duration. Talking machines: Theories, models, and designs pp. 211–224 (1992)
3. He, K., Zhang, X., Ren, S., Sun, J.: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (December 2015)
4. Jiang, Y., Hu, Y., Liu, L., Wu, H., Wang, Z., Ai, Y., Ling, Z., Dai, L.: The USTC System for Blizzard Challenge 2019. In: Blizzard Challenge Workshop 2019. Vienna, Austria (September 2019)
5. Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., Kavukcuoglu, K.: Efficient Neural Audio Synthesis. arXiv e-prints arXiv:1802.08435 (Feb 2018)
6. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Klatt, D.H.: Interaction between two factors that influence vowel duration. The Journal of the Acoustical Society of America **54**(4), 1102–1104 (1973)
8. Louw, J.A.: Neural speech synthesis for resource-scarce languages. In: Barnard, E., Davel, M. (eds.) Proceedings of the South African Forum for Artificial Intelligence Research. pp. 103–116. Cape Town, South Africa (December 2019)
9. Louw, J.A., Moodley, A., Govender, A.: The Speect text-to-speech entry for the Blizzard Challenge 2016. In: Blizzard Challenge Workshop 2016. Cupertino, United States of America (September 2016)
10. Louw, J.A., van Niekerk, D.R., Schlünz, G.: Introducing the Speect speech synthesis platform. In: Blizzard Challenge Workshop 2010. Kyoto, Japan (September 2010)
11. Morais, E., Violaro, F.: Exploratory analysis of linguistic data based on genetic algorithm for robust modeling of the segmental duration of speech. In: Ninth European Conference on Speech Communication and Technology (2005)
12. van Niekerk, D., de Waal, A., Schlünz, G.: Lwazi II Afrikaans TTS Corpus. `https://repo.sadilar.org/handle/20.500.12185/443` (November 2015), ISLRN: 570-884-577-153-6
13. Ren, Y., Hu, C., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.: FastSpeech 2: Fast and High-Quality End-to-End Text-to-Speech. arXiv preprint arXiv:2006.04558 (2020)
14. Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.: Fastspeech: Fast, robust and controllable text to speech. In: Advances in Neural Information Processing Systems. pp. 3171–3180 (2019)

15. Riley, M.D.: Tree-based modelling for speech synthesis. In: The ESCA Workshop on Speech Synthesis. pp. 229–232 (1991)
16. Shen, J., Pang, R., Weiss, R.J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R.A., Agiomyrgiannakis, Y., Wu, Y.: Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. arXiv e-prints arXiv:1712.05884 (Dec 2017)
17. Silverman, K., Beckman, M.E., Pitrelli, J., Ostendorf, M., Wightman, C., Price, P., Pierrehumbert, J., Hirschberg, J.: ToBI: a standard for labeling English prosody. In: Proceedings of the Second International Conference on Spoken Language Processing (ICSLP). pp. 867–870. Alberta, Canada (October 1992)
18. Sotelo, J., Mehri, S., Kumar, K., Santos, J.F., Kastner, K., Courville, A., Bengio, Y.: Char2wav: End-to-end speech synthesis. arXiv preprint arXiv:1609.03499 (2017)
19. Tachibana, H., Uenoyama, K., Aihara, S.: Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention. arXiv e-prints arXiv:1710.08969 (Oct 2017)
20. Taylor, P.: Text-to-Speech Synthesis. Cambridge University Press (2009)
21. Tokuda, K., Nankaku, Y., Toda, T., Zen, H., Yamagishi, J., Oura, K.: Speech synthesis based on hidden Markov models. Proceedings of the IEEE **101**(5), 1234–1252 (2013)
22. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K.: WaveNet: A Generative Model for Raw Audio. arXiv e-prints arXiv:1609.03499 (Sep 2016)
23. Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R.J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q., Agiomyrgiannakis, Y., Clark, R., Saurous, R.A.: Tacotron: Towards End-to-End Speech Synthesis. arXiv e-prints arXiv:1703.10135 (Mar 2017)
24. Watts, O., Henter, G.E., Fong, J., Valentini-Botinhao, C.: Where do the improvements come from in sequence-to-sequence neural TTS? In: 10th ISCA Speech Synthesis Workshop. ISCA, Vienna, Austria (September 2019)
25. Wei, X., Hunt, M., Skilling, A.: Neural Network-Based Modeling of Phonetic Durations. arXiv preprint arXiv:1909.03030 (2019)
26. Wu, Z., Watts, O., King, S.: Merlin: An Open Source Neural Network Speech Synthesis System. In: SSW. pp. 202–207 (2016)
27. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., et al.: The HTK book. Cambridge University Engineering Department **3**, 175 (2002)
28. Zen, H., Tokuda, K., Masuko, T., Kobayasih, T., Kitamura, T.: A Hidden Semi-Markov Model-Based Speech Synthesis System. IEICE Transactions on Information and Systems **E90-D**(5), 825–834 (May 2007)
29. Zen, H., Senior, A.: Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 3844–3848. IEEE (2014)
30. Zhu, X., Zhang, Y., Yang, S., Xue, L., Xie, L.: Pre-alignment guided attention for improving training efficiency and model stability in end-to-end speech synthesis. IEEE Access **7**, 65955–65964 (2019)