INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Idlak Tangle: An Open Source Kaldi Based Parametric Speech Synthesiser based on DNN

Blaise Potard[1,3], *Matthew P. Aylett*[1,2], *David A. Baude*[1], *Petr Motlicek*[3]

[1]CereProc Ltd., United Kingdom
[2]The Centre for Speech Technology Research, University of Edinburgh, United Kingdom
[3]The Idiap Research Institute, Martigny, Switzerland

Abstract

This paper presents a text to speech (TTS) extension to Kaldi - a liberally licensed open source speech recognition system. The system, Idlak Tangle, uses recent deep neural network (DNN) methods for modelling speech, the Idlak XML based text processing system as the front end, and a newly released open source mixed excitation MLSA vocoder included in Idlak. The system has none of the licensing restrictions of current freely available HMM style systems, such as the HTS toolkit. To date no alternative open source DNN systems are available. Tangle combines the Idlak front-end and vocoder, with two DNNs modelling respectively the units duration and acoustic parameters, providing a fully functional end-to-end TTS system.

Experimental results using the freely available SLT speaker from CMU ARCTIC, reveal that the speech output is rated in a MUSHRA test as significantly more natural than the output of HTS-demo, the only other free to download HMM system available with no commercially restricted or proprietary IP. The tools, audio database and recipe required to reproduce the results presented in these paper are fully available online.
Index Terms: Speech synthesis, Kaldi, Idlak, HTS, DNN

1. Introduction

Statistical parametric speech synthesis based on Hidden Markov Models (HMMs) has become a common method for generating highly intelligible, flexible speech output. The dominant system, HTS [1], has been developed for over a decade, and led the way in developing parametric synthesis approaches and algorithms.

More recently, spurred on by the success of Deep Neural Networks (DNNs) in speech recognition [2], significant research has been carried out investigating the use of DNNs in parametric speech synthesis [3]. One approach is the use of DNNs to replace Gaussian mixture models (GMMs) associated with leaf nodes of decision trees. Ling *et al.* [4], using restricted Boltzmann machines (RBMs) claim a neural network approach is better at learning spectral detail than GMMs and decision trees, resulting in better quality speech output. Furthermore, DNNs offer the ability to model high-dimensional acoustic parameters [5].

The work presented here does not attempt to extend the state-of-the-art in DNN synthesis. Rather, the objective is to offer both a simple, understandable baseline for the community that can be compared against more complex systems, and a practical recipe that offers a real opportunity for open source development of Text-To-Speech (TTS) in under resourced languages without requiring access to bespoke and propriety techniques. Furthermore, by embedding the system within Idlak-Kaldi, we believe that parallel work in speech recognition can be more readily harnessed to improve TTS, and that by offering a reproducible system with a significant improvement in quality from the currently unrestricted HMM-based solution, and without license restrictions, we can greatly encourage advances in the field.

Idlak is a project to build an end-to-end parametric synthesis system within Kaldi [6], a liberally licensed Automatic Speech Recognition (ASR) toolkit. As part of Idlak, a front-end that generates full-context models compatible with HTS has been developed [7]. This front-end performed well in an evaluation against Festival, a standard front-end used by HTS. In this paper, we explore the use of one of Kaldi's DNN frameworks as an alternative to the HTS/HTK system. We have called this end-to-end TTS-DNN system *Tangle*. As with Kaldi, Idlak and Tangle are both liberally licensed.

The system we present first uses Kaldi to carry out a phone alignment on a single-speaker corpus. This alignment is then used to train two cascading DNNs: one for predicting unit durations, and one for predicting acoustic output. Also incorporated are analysis and synthesis tools to perform MLSA vocoding with mixed excitation [8] and a simple recipe to encourage other research groups to reproduce our results. All the necessary code can be downloaded from the Kaldi-Idlak repository[1] allowing our results to be reproduced. Tangle only depends on tools that use either BSD (SPTK, expat, PCRE), Apache (Kaldi, openfst), or MIT (pugixml) licenses, allowing the use of Tangle for commercial or academic applications. Tangle is itself released under the Apache license.

The primary motivation for our work can be summarised as follows:

1. It is part of a long-term goal to produce a Kaldi-based end-to-end parametric speech synthesis system. HTS suffers from licensing restrictions that prevent a standard open-source model. In addition, many new approaches in ASR are already implemented within Kaldi, such as sophisticated DNNs [9] and sub-space modelling [10]. This would allow current and future ASR developments to be directly incorporated into speech synthesis as they become available and vice versa.
2. High quality vocoders are a requirement for a good sounding parametric TTS system; however most of the available ones are either low quality or suffer from licensing restrictions that make them unsuitable to be included directly into an open source project with a liberal

[1]Currently the Idlak branch of Kaldi can be installed with `git clone https://github.com/bpotard/idlak.git`

http://dx.doi.org/10.21437/Interspeech.2016-1188

license. By re-implementing state-of-the-art vocoding techniques into Kaldi, we hope to bridge that gap and offer the first free high quality parametric Text-to-Speech system.

3. By making our system openly available, together with the tests we describe, we offer a useful test harness and a better sounding baseline than HTS-demo to the community.

In the following sections we discuss the choice of our HMM based (HTS) parametric speech synthesis baseline, a description of the DNN modelling process, and the speech synthesis process. We continue by carrying out a listening test comparing a set of different synthesisers, present our evaluation and conclude by discussing the choices made in our design, and potential future work.

2. Using HTS-demo as an Idlak baseline

Idlak supplies a bash script to download and build the publicly available HTS-demo and its dependencies for Linux based systems for comparisons. The full-context models for both training and running the system are then replaced as detailed in [7] and the Idlak documentation. Subsequently, standard HTS-demo training is carried out, followed by the equally standard synthesis of the beginning of Lewis Carroll's "Alice's Adventures in Wonderland"[2] using HTS-engine.

HTS-demo was chosen as a baseline, not because it is the best HMM system available (there are many better and more sophisticated systems presented in previous research), but because it is the only system we could source that did not require proprietary audio databases, proprietary lexicons, or proprietary signal analysis (e.g. STRAIGHT [11]). Kaldi itself was partly the result of the difficulties of adapting HTK for research work because it has license restrictions that considerably limits its use in conventional open source projects. These same restrictions are present in HTS-demo, which relies on HTK to build HMM models and trees together with a patch. However, the models created by the training process can then be freely distributed and some can be used using free software tools.

In this paper we will compare Tangle to the output of HTS-demo (v. 2.3 alpha) using the SPTK toolkit (v. 3.6) for acoustic analysis, and HTS engine (v. 1.07) for synthesis. The speech database used is the standard HTS-demo database, i.e. CMU ARCTIC speaker SLT, upsampled to 48 kHz.

3. DNN-based duration and acoustic models

3.1. Generalities

A collection of tools and scripts were added to the Idlak toolkit to allow the training of DNNs suitable for TTS. The internal structures, training procedures and methods were derived from the *nnetbin* DNN variant of Kaldi.

The Idlak front-end analyses and normalises input text, then generates a rich phonetic and contextual representation from it, a.k.a "full labels". The Idlak text processing modules each operate on an XML marked-up stream of text. Each module will typically add structure to the XML and may be dependent on structure added by previous modules. Figure 1, shows the current modules that form idlaktxp. See [7] for more details.

[2]The full text is available from Project Gutenberg `http://www.gutenberg.org/ebooks/11`

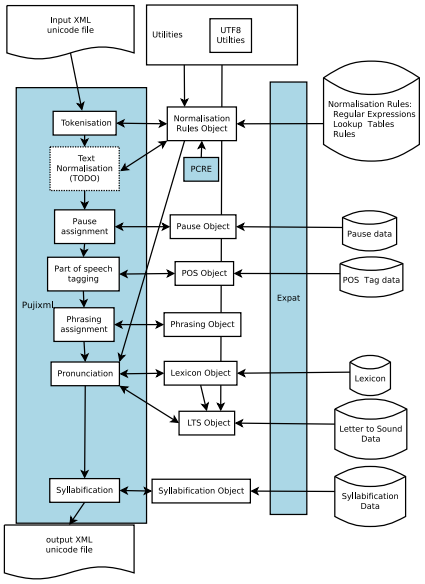

Figure 1: *Idlak text processing system (idlaktxp). The system comprises of a set of modules operating on XML input and producing further tagged XML.*

In tangle, two deep neural networks need to be trained: a *Duration Model* DNN (DM-DNN) that will predict the durations of both phones and HMM states from input phone labels, and an *Acoustic Model* DNN (AM-DNN) that will predict an acoustic sequence from a sequence of acoustic labels.
Figure 2 summarises the training procedure.

The AM-DNN training requires a frame-level mapping between input labels and acoustic features; therefore the unit-level labels have to be sampled so that we have an input label per acoustic frame. Based on our previous work [12], we chose to add 2 numerical values to the full context labels, respectively coding for the frame position within the current HMM state, and the position within the current phone. We treated the state identity as a numerical value rather than a categorical feature.

As this system is more intended to be used as a light-weight baseline rather than a state-of-the-art system, the DNNs were built using relatively modest numbers of hidden layers (3), and nodes (100 and 700, respectively, for the duration and acoustic models) in each layer. Each layer comprised an affine component followed by a sigmoid activation function.

The input data (label) was further normalized for each component to be of zero mean and unit variance. To reduce the issues linked to frame-by-frame independence, we spliced together 11 input frames (5 back, 5 front), which gave us input dimensions of respectively 4125 and 4169 for duration / acoustic DNNs.

The output data (duration or acoustic) was normalized globally so that each output component had values between 0.01 and 0.99; the output activation function was a sigmoid.

Unlike other approaches (such as Zen [13] or Qian [14]), we did not remove silent frames from the training, as it was not found to be necessary for synthesis quality. The training procedure was standard: we used a stochastic gradient descent based on back propagation. The minimisation criterion was the Mean Square Error (MSE). We did not use dropout. The training was run on a *training* set, and we used a *development* set for cross-validation.

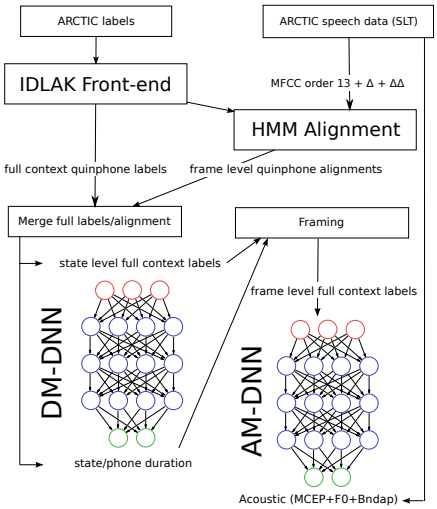

Figure 2: *Tangle DNN training architecture*

3.2. Forced alignment procedure

A forced alignment procedure performed on the full database was used to align the full context labels with the acoustic data, using standard tools from the Kaldi toolkit.

The models for the alignment were trained on the training plus development sets, and state-level labels force-aligned to acoustic frame boundaries were generated for the training and development sets. The models used were 5-states left-right HMMs with multiple Gaussians, 3230 tied HMM states and about 50k Gaussians were used. The acoustic features used for alignment were order 13 MFCC with first and second order derivatives.

3.3. Duration modelling DNN

We trained a first DNN to learn a mapping between full label information and the respective durations of states and phones.

The input of this DNN is the full label mapped to numerical features, the output is the respective duration (as a number of frames) of the "units" the label belong to, which we limited to phone and HMM state, as extracted from the forced alignment.

In case some states had been skipped in the alignment, input for the skipped states were added with an output duration of 0 for the state.

3.4. Acoustic modelling DNN

The input of this DNN needs to have the same sampling rate as the acoustic data, so the full labels with state and phone duration need to be oversampled. In practice, we duplicate as many state-level labels as needed based on the output predicted by the DM-DNN. The input frames within a state are then made distinct by appending quantized positions, respectively within the current state, and within the current phone. The positions within the state were restricted to 5 distinct values, while the positions within the phone were restricted to 10 distinct values.

The acoustic features contained 2 values for modelling the periodic excitation (continuous F0 and voicing probability), 25 values for aperiodic excitation (Bark-scale band aperiodicity), and 60 values for modelling the filter. First and second order derivatives of all these features were also modelled, for a total output vector size of 261.

3.5. DNN synthesis

In practice, the synthesis procedure works as follows:

1. The Idlak front-end turns an input text into "full labels", a rich phonetic and contextual representation of it.
2. The full labels are transformed into an input suitable for the DNN by mapping all features to numerical values.
3. The durations within each phone and within each HMM state are predicted using the *Duration model* DNN.
4. The input to the *Acoustic model* DNN is generated by creating input label frames for each acoustic frames desired, i.e. input labels for each HMM states are duplicated as needed to match the predicted durations. The quantized positions within the phone and the HMM states are appended to the input labels.
5. The raw acoustic features and their derivatives are predicted using the *Acoustic model* DNN.
6. The acoustic features trajectories are smoothed using the MLPG algorithm.
7. An excitation signal is built using the voicing and band aperiodicity information.

During synthesis, the "full" labels generated on input text by the Idlak front-end are converted to numerical values, then output durations are calculated by forward propagation in the DM-DNN. These values are then post-processed for consistency, so that the sum of the states durations within a given phone is equal to the phone duration.

By combining input labels and durations together, we can then generate valid input for the acoustic model DNN. The full labels with state and phone duration appended are oversampled as described in the previous sub-section, and then forward propagated in the AM-DNN.

This generates sequences of acoustic features with their derivatives; these sequences are post-processed using the MLPG algorithm to generate a smooth sequence of acoustic features. These features are then fed to a mixed excitation MLSA synthesizer [8] to generate the audio output.

The analysis and synthesis tools used have been integrated to Idlak.

4. Experiment

For a fair comparison to the HTS-demo system, we used the same audio database for both systems: the CMU ARCTIC database speaker SLT. The training set consisted of 1132 audio files, encoded in mono PCM wave format, with a sampling rate of 48kHz (upsampled from 32kHz), totalling 47:01 minutes once start and end silences had been trimmed.

	HTS-demo	Tangle DNN
Filter	MCEP ord. 60	MCEP ord. 60
Periodic exc.	Discontinuous	Continuous
Aperiodic exc.	None	Band aperiod. (ord. 25)

Table 1: *Acoustic parametrisation.*

The tools supplied with Idlak to build an HTS-demo with the Idlak front-end were used as detailed in [7], followed by synthesis of the beginning of Lewis Carroll's "Alice's Adventures in Wonderland" using HTS engine. Note that apart from the front-end used to generate the labels, this setup is strictly equivalent to the default settings of HTS-demo. Note also that training the HTS models requires the proprietary HTK toolkit, which requires registration, but the synthesis procedure can be performed with the `hts_engine` tool, which is distributed as free software.

The acoustic parametrisation used for both Tangle and HTS-demo is summarised in Table 1. The MCEP coeffi-

cients were extracted in both cases using SPTK MCEP extraction tool with $\alpha = 0.55$. The periodic excitation was extracted using respectively SPTK's `pitch` and Kaldi's `compute-kaldi-pitch-feats`, and the aperiodic energy was extracted using Idlak's `compute-aperiodic-feats`.

Note that the main purpose of this evaluation is to compare open source parametric synthesis systems that can be used for any purpose (including commercial use), therefore the proprietary vocoder STRAIGHT was not used to build the HTS-demo voice.

For reference purposes, a unit selection voice built by a proprietary system and a commercial grade HTS voice were provided by CereProc Ltd, created on the same audio database.

As the speech database is very small, the unit selection voice was not expected to perform significantly better than the parametric systems [15].

5. Evaluation

15 expert listeners completed a MUSHRA-like preference test [16] on 12 output phrases selected to cover different phrase lengths, where the listeners were tasked in rating between 0 and 100 the naturalness of the outputs generated by each of the 4 systems. Note that the test had neither reference nor anchor, as there are no original recordings of these samples by the target speaker, and none of the system was expected to be consistently better or worse than all the others.

For statistical analysis, opinion scores were averaged across subjects to produce an average score for each phrase[3]. A repeated-measures ANOVA was carried out by phrase, with four conditions: HTS-demo, HTS commercial, Idlak Tangle, Unit Selection. Results showed a significant difference between groups ($F(3, 33) = 29.821, p < 0.001$), pairwise comparison of the means using the least significant difference (LSD) procedure with Bonferroni correction showed a significant difference ($p < 0.025$) between all means except between the commercial HTS system and Idlak Tangle. The unit selection system performed best but with a wider variance, and Idlak Tangle DNN system consistently outperformed the HTS-demo baseline. However it was neither better nor worse than the proprietary HMM system.

6. Discussion

These example voices were built from the freely available ARCTIC SLT voice. With 47 minutes of data this is a small corpus for TTS voice building by today's standards. Given the small size of the database the unit selection system performed surprisingly well. Results in Blizzard [15] challenges have generally shown unit selection voices to be below parametric quality for databases on this size.

Previous work on DNN approaches to synthesis have typically used larger databases (e.g. Zen *et al.* [13] 30 hours, Wu and King 2400 utterances). The results here show that with the right architecture, a DNN solution can also outperform, or match, an HMM system with a small corpora. This is especially important for less resourced languages where the expense of recording many hours of data can be a barrier to development.

Readers are encouraged to listen to the samples at `http://homepages.inf.ed.ac.uk/matthewa/interspeech2016DNN` to gauge the quality of both the HMM baseline and the Tangle DNN system. The HTS Demo output does not use mixed excitation. Within HTS-based systems, mixed excitation is typically driven with band aperiodic energy parameters produced using the restricted license STRAIGHT [11] system. Hence the lack of mixed excitation in the HTS-demo output which leads to a strong sense of audio buzz in voiced regions. An important contribution from this work is a open source method of determining aperiodic band energy for use in more freely licensed systems.

[3]By assuming the discrete opinion scores are independent and identically distributed samples, we are able to use the central limit theorem to regard the means as being drawn from an approximately Gaussian distribution [17].

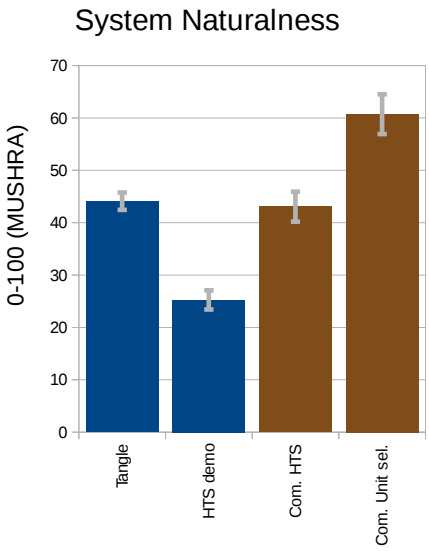

Figure 3: *Mean opinion score of the four systems: 1. with Idlak Tangle, 2. with HTS-demo Baseline, 3. with commercial HTS style system, 4. with commercial Unit Selection system. Error bars show standard error. All means except for DNN and commercial HTS are significantly different ($p < 0.025$).*

The voice quality produced by Tangle is not buzzy, but does exhibit the dull and muffled quality associated with early HMM systems which did not use global variance to increase the variance in trajectory modelling. In this early system no attempt has been made to use global variance or variance scaling to increase variability of the speech output. Furthermore, the vocoder used does not implement phase randomisation. Future work intends to improve this part of the released vocoder.

7. Conclusions

The DNN Tangle system presented here is using a simple, open framework. Compared to the HTS-based system, the architecture and the licensing situation is simple and allows liberal use of the system within both commercial and academic environments. The performance of Tangle is significantly better than the baseline HTS-demo parametric system. Tangle is to our knowledge the first DNN-based parametric synthesis system with no usage restriction, however this is by no means the current state of the art. We look forward to other research groups comparing Tangle to their own systems and contributing to the Idlak Tangle open source project.

8. Acknowledgements

This work was funded by the Eurostars Programme - powered by Eurostars and the European Community - under the project "D-Box: A generic dialogue box for multi-lingual conversational applications", and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 645378 (Aria VALUSPA).

9. References

[1] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. Black, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. SSW6*, 2007, pp. 294–299.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. M. Meng, and L. Deng, "Deep learning for acoustic modeling in parametric speech generation: A systematic review of existing techniques and future trends," *Signal Processing Magazine, IEEE*, vol. 32, no. 3, pp. 35–52, 2015.

[4] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted boltzmann machines and deep belief networks for statistical parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 10, pp. 2129–2139, 2013.

[5] Z. Wu and S. King, "Improving trajectory modelling for dnn-based speech synthesis by using stacked bottleneck features and minimum trajectory error training," *arXiv preprint arXiv:1602.06727*, 2016.

[6] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," *Proc. IEEE ASRU*, 2011.

[7] M. P. Aylett, R. Dall, A. Ghoshal, G. E. Henter, and T. Merritt, "A flexible front-end for HTS," in *Proc. Interspeech*, 2014, pp. 1283–1287.

[8] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Mixed excitation for HMM-based speech Synthesis," in *Proceedings of Eurospeech*, 2001, pp. 2259–2262.

[9] K. Veselỳ, Karel, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Proc. Interspeech*, 2013, pp. 2345–2349.

[10] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "The subspace Gaussian mixture model—a structured model for speech recognition," *Comput. Speech Lang.*, vol. 25, no. 2, pp. 404–439, 2011.

[11] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoust. Sci. Technol.*, vol. 27, no. 6, pp. 349–353, 2006.

[12] A. Lazaridis, B. Potard, and P. N. Garner, "DNN-based Speech Synthesis: Importance of input features and training data," in *International Conference on Speech and Computer , SPECOM*, ser. Lecture Notes in Computer Science, A. Ronzhin, R. Potapova, and N. Fakotakis, Eds. Springer Berlin Heidelberg, 2015, vol. 9319, pp. 193–200.

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

[14] Y. Qian, Y. Fan, W. Hu, and F. Soong, "On the training aspects of deep neural network (dnn) for parametric tts synthesis," in *Proc. of ICASSP*, 2014, pp. 3829–3833.

[15] M. P. Aylett, C. J. Pidcock, and M. E. Fraser, "The cerevoice blizzard entry 2006: A prototype database unit selection engine," in *In Proc. BLIZZARD Challenge*, 2006.

[16] *Method for the Subjective Assessment of Intermediate Quality Level of Coding Systems*, International Telecommunications Union Std. ITU-R Rec. BS.1534-1, 2003.

[17] H. N. Boone and D. A. Boone, "Analyzing Likert data," *J. Extension*, vol. 50, no. 2, pp. 1–5, 2012.