*INTERSPEECH 2017*
August 20–24, 2017, Stockholm, Sweden

# Google's Next-Generation Real-Time Unit-Selection Synthesizer using Sequence-To-Sequence LSTM-based Autoencoders

*Vincent Wan, Yannis Agiomyrgiannakis, Hanna Silen, Jakub Vit*

Google, London, UK
{vwan,agios,silen}@google.com, jvit@kky.zcu.cz

## Abstract

A neural network model that significant improves unit-selection-based Text-To-Speech synthesis is presented. The model employs a sequence-to-sequence LSTM-based autoencoder that compresses the acoustic and linguistic features of each unit to a fixed-size vector referred to as an *embedding*. Unit-selection is facilitated by formulating the target cost as an $L_2$ distance in the embedding space. In open-domain speech synthesis the method achieves a $0.2$ improvement in the MOS, while for limited-domain it reaches the cap of $4.5$ MOS. Furthermore, the new TTS system halves the gap between the previous unit-selection system and WaveNet in terms of quality while retaining low computational cost and latency.

**Index Terms**: text-to-speech synthesis, LSTM, unit-selection

## 1. Introduction

Generative Text-To-Speech (TTS) has improved over the past few years and challenges traditional unit-selection approaches [1,2] both at the low-end and the high-end parts of the market where the computational resources are scarce and excessive, respectively. At the low-end market, such as TTS running on mobile devices, unit-selection is challenged by statistical parametric speech synthesis (SPSS) [3, 4], while it is challenged by advanced approaches like WaveNet [5] at the high-end market. Although SPSS-based TTS is sometimes preferred over unit-selection-based TTS for a mildly-curated speech corpus [4], it is not preferred over unit-selection for voices based on highly-curated speech corpora. Meanwhile, WaveNet is not fast enough to be used in practice for the average use-case. Yet the ability of unit-selection to yield studio-level quality for limited-domain TTS remains largely unchallenged. This creates a time window where unit-selection methods can still deliver higher quality to the market.

Improving unit-selection TTS using neural networks has so far yielded results [6–8] that are not as impressive as those obtained for SPSS [3, 4, 9–13] when the transition from hidden Markov models (HMMs) to neural networks was made.

The approach in [6] is computationally expensive. It runs an SPSS network similar to [4] with a bidirectional long short-term memory (bLSTM) network to predict the vocoder parameter sequence of each unit. The predicted parameter sequence is compared to the vocoder parameter sequence of the units in the database by various metrics to determine the target cost.

A more efficient approach is to construct a fixed-size representation of the variable-size audio units, hereafter referred to as a (unit-level) *embedding*. Both [7, 8] approaches take frame-level embeddings of linguistic and acoustic information from the intermediate layers of a deep neural network (DNN) [7] or a long short-term memory (LSTM) [8] network and use them to construct a unit-level embedding. In [7], unit-level embeddings are made by segmenting each unit in to four parts and taking the short-term statistics (means, variances) of each part. In [8], the frame-level embeddings are sampled at fixed-points on a normalized time axis. In both cases, the unit-level embeddings are constructed heuristically rather than being learned directly. From a modelling perspective, such heuristic approaches limit the effectiveness of the embedding both in terms of compactness (yields larger unit-embeddings) as well as reconstruction error (information is lost both through sampling or taking short-term statistics).

This paper presents a significant improvement to unit-selection technologies when replacing the HMM with a sequence-to-sequence LSTM-based autoencoder. In particular, a network with a temporal bottleneck layer represents each unit of the database with a single embedding. An embedding should satisfy some basic conditions for it to be useful for unit-selection:

1. it can encode variable-length audio to a fixed-length vector representation;
2. it must represent the acoustics;
3. it must be possible to infer the embedding from the linguistic features;
4. the metric of the embedding space should be meaningful; similar sounding units should be close together while units that are different should be far apart.

Work by [14] for parametric speech synthesis employs sequence-to-sequence autoencoders to compress the frame-level *acoustic* sequence onto a unit-level *acoustic embedding*. During synthesis they use a second network that is trained separately to infer the acoustic embeddings from the linguistic information. Our model is similar to [14] but besides being applied to unit-selection it has the key difference that the autoencoder network and the text-to-embedding network are trained jointly and that we employ an LSTM-based recurrent neural network (RNN) instead of a simple RNN. Unit-selection is facilitated by formulating the target cost as the $L_2$ distance in the embedding space. The use of $L_2$ instead of Kullback-Leibler distance allows us to reduce the computational cost significantly by re-casting preselection as a $k$-nearest neighbor problem.

Section 2 describes how the unit embeddings in a TTS database are learned automatically and deployed in a unit-selection TTS system. Sections 3 and 4 describe the experiments and results respectively.

## 2. Using LSTMs for unit preselection

Both acoustic (speech) and linguistic (text) features are available during training but only the linguistic features are present

Jakub Vit is a PhD. student at the Department of Cybernetics, University of West Bohemia, Pilsen, Czech Republic, and performed this work during his internship at Google.

http://dx.doi.org/10.21437/Interspeech.2017-1107

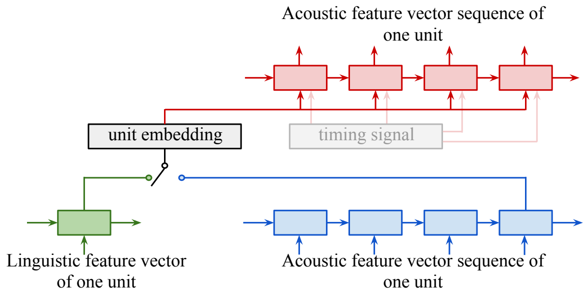
Acoustic feature vector sequence of one unit
unit embedding
timing signal
Linguistic feature vector of one unit
Acoustic feature vector sequence of one unit

Figure 1: *Sequence to sequence autoencoder network for learning embeddings representing phone or diphone sized units. It consists of a linguistic encoder (green) and an acoustic encoder (blue) connecting to an acoustic decoder (red) via a switch. The decoder also receives a timing signal indicating how far it is through decoding the current unit [13].*

at run-time. The first challenge is to design a network that is able to exploit both at the input of the network during training but still works correctly at run-time without acoustic features. This is desirable for unit-selection because it is important that the embedding represents the acoustic content of the unit: since the linguistic features alone are insufficient to describe the full variability that exists in each unit, without the acoustics it is likely that the network will learn a smoothed or average embedding. Furthermore, if the learned embeddings are unconstrained then they can vary hugely between different training sessions depending upon the network's initialization. Such variability can pose problems for unit-selection when the target cost, estimated as the $L_2$ distance between embeddings (section 2.4), is combined with join costs in the Viterbi search for the best path.

### 2.1. Learning the embeddings

The topology of the proposed network is shown in figure 1. This approach is similar to multimodal embeddings for learning joint text/image embeddings [15, 16]. Embeddings are learned using a sequence to sequence autoencoder network consisting of LSTM units. The network consists of two encoders: The first encodes the linguistic sequence, which consists of a single feature vector for each (phone- or diphone-sized) unit. It is a multilayer recurrent LSTM network that reads one input linguistic feature vector and outputs one embedding vector for every unit. The second encodes the acoustic sequence of each unit. It too is a recurrent multilayer LSTM network. Its input is the sequence of parameterized acoustic features of a complete unit and it outputs one embedding vector upon seeing the final vector of the input sequence. This is the aforementioned *temporal bottleneck* where information from multiple time frames is squeezed into a single low dimensional vector representation.

The embedding outputs of the two encoders are the same size. A switch is inserted so that the decoder may be connected to either the acoustic or the linguistic encoder. During training the switch is set randomly for each unit according to some fixed probability.

The decoder is trained to estimate the acoustic parameters of the speech given the embedding from the decoder to which is it connected. Its topology is similar to [17] with the input composed of the embedding vector duplicated enough times to match the number of frames in the unit plus a coarse coding timing signal is appended to each frame, which tells the network how far it is through the unit.

The network is trained using back-propagation through time with stochastic gradient descent and a squared error cost is used at the output of the decoder. Since the output of the encoder is only taken at the end of a unit, error back-propagation is truncated at unit boundaries: truncating on a fixed number of frames may result in weight updates that do not account for the start of a unit. To further encourage the encoders to generate the same embedding an additional term is added to the cost function to minimize the squared error between the embeddings produced by the two encoders. This joint training allows both acoustic and linguistic information to influence the embedding while creating a space that may be mapped to when given only linguistic information. In the set up described in [14] linguistic information is not incorporated into the embedding because it is learned entirely by the autoencoder: The linguistic encoder is trained separately after the acoustic encoder has been finalized.

### 2.2. Partitioned embeddings

One feature of unit-selection systems is the ability to weight the relative importance of the different information streams, spectrum, aperiodicity, $F_0$, voicing and duration. Using a single decoder will result in an embedding that encodes all these streams into the embedding making it impossible to reweight the streams. So that reweighting may be achieved, the embedding is partitioned into separate streams and each partition is connected to its own decoder that is solely responsible for predicting the features of that stream.

### 2.3. Isometric embeddings

We introduce isometric embeddings as an additional constraint so that $L_2$ distances within the embedding space, firstly, become direct estimates of the acoustic distance between units, and secondly, are more consistent across independent network training runs. This tries to give the $L_2$ distance between embeddings a meaningful interpretation since we use it as the target cost and combine it with join costs in unit-selection.

Define the dynamic time warping (DTW) distance between pairs of units as the sum over the $L_2$ distances between pairs of frames in the acoustic space aligned using the DTW algorithm. We add a term to the network's cost function such that the $L_2$ distance between the embedding representations of two units is proportional to the corresponding DTW distance. This is implemented by training the network using batch sizes greater than one. Phones from different sentences in the mini-batch are aligned using DTW to yield a matrix of DTW distances. The corresponding $L_2$ distance matrix is computed between the phones' embeddings. The difference between these two matrices is added to the network's cost function for minimization.

### 2.4. Nearest neighbour preselection

When building the voice the embeddings of every unit in the voice training data are saved in a database. At run-time, the linguistic features of the target sentence are fed through the linguistic encoder to get the corresponding sequence of *target embeddings*. For each of these target embeddings $k$-nearest units are preselected from the database. These preselected units are placed into a lattice and a Viterbi search is performed to find the best sequence of units that minimizes the overall target and join costs. The target cost is calculated as the $L_2$ distance from the target embedding vector predicted by the linguistic encoder to the unit's embedding vector stored in the database. The join cost is the same one used in [2].

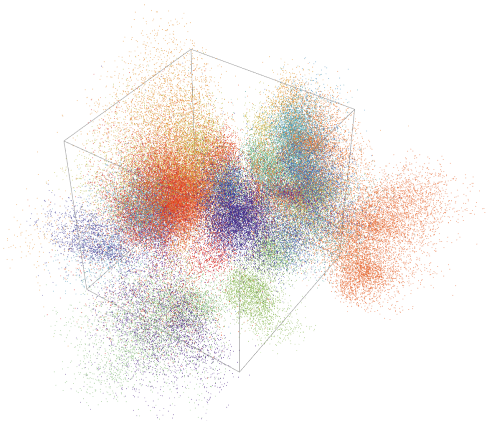

Figure 2: *Visualizing the embedding space: The network was set up to learn 3-dimensional embeddings of phone-based units. The coloured groups are clusters of the same phone.*

# 3. Experimental setup

## 3.1. Training data

The training data consists of around 40,000 sentences recorded from a single American English speaker in carefully controlled studio conditions. For the experiments described in this paper, the audio was down-sampled to 22.05 kHz. The speech was parameterized as 40 Mel-scaled cepstral coefficients, 7 band aperiodicities, $\log -F_0$ and a boolean indicating voicing. 400 sentences, chosen at random, were held out as a development set to check that the networks did not over-train.

## 3.2. Baseline

The baseline system is described in [2]. It uses HMMs and KL divergence to preselect diphones which are placed into a lattice. A Viterbi search is used find the best sequence that minimizes the target and join costs.

## 3.3. Subjective tests

Subjective evaluation of unit-selection systems is particularly sensitive to the selection of test-set utterances because the MOS of each utterance depends on how well the utterance matches the statistics of the audio corpus. To mitigate this, we strengthen the standard testing practice [2, 3] as follows: First, we shift the statistical power of the listening test towards utterance coverage by having only one rating per utterance and 1,600 utterances. Second, we sample the test utterances directly from anonymized TTS logs using uniform sampling on the logarithmic frequency of the utterances. This ensures that the test-set is representative of the actual user experience and that the MOS results are not biased towards the head of the Zipf-like distribution of the utterances.

All subjective evaluations are made using the Mean-Opinion-Score (MOS) of naturalness, a supervised high-quality crowdsourced system and the standard 5-point Likert scale [3]. Due to space-limitations we present results in two distinctive domains: *limited-domain* (*Maps*) and *open-domain* (*WebAnswers*), obtained using 1,600 utterances per domain, one rating per utterance and more than a thousand raters.

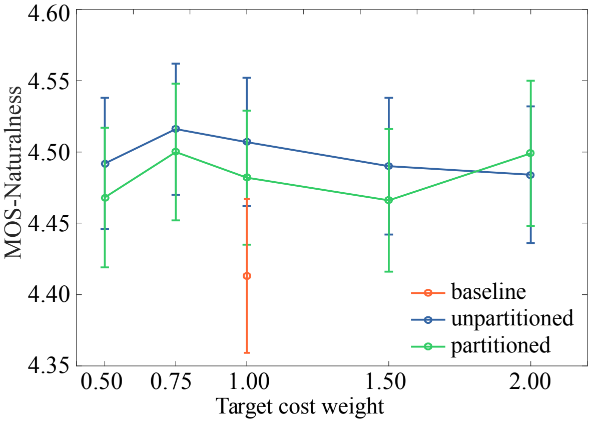

Figure 3: *Weight tuning for limited-domain TTS (Maps).*

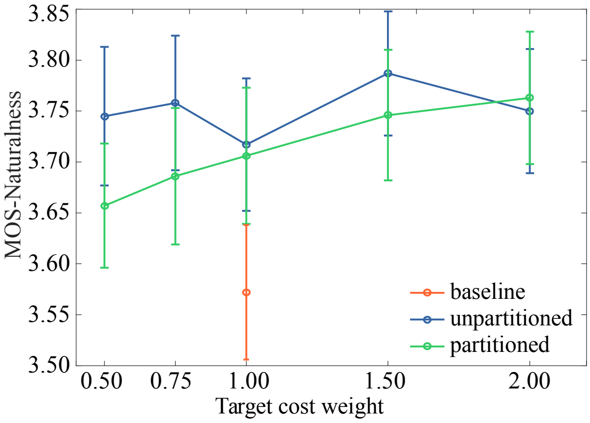

Figure 4: *Weight tuning for open-domain TTS (WebAnswers).*

# 4. Results

Low-order embeddings are surprisingly informative. We are able to reconstruct highly intelligible medium quality parametric speech [3] with only 2 or 3 parameters per phone, rendering the proposed method suitable for ultra-low-bit-rate speech coding. Further, the embeddings are meaningful in the sense that adjacent points in the embedding space correspond to phonemes that have identical or very similar contexts. Thus, the proposed method is an excellent way to visualize speech. Figure 2 shows a 3D scatter plot illustrating the phoneme clusters obtained from a network trained to produce a 3-dimensional embedding space.

Preliminary informal listening tests showed that phoneme-based embeddings perform better than diphone-based ones. This can be attributed to the fact that a phone is a much more compact abstraction of a unit than a diphone. In other words, the lower cardinality of the phone set improves the efficiency of the corresponding embedding.

Our first experiment is geared towards tuning two different systems: *unpartitioned* and *partitioned*. The two systems differ only on whether the information streams that describe unit acoustics (spectra, aperiodicity, $\log -F_0$, voicing) are embedded jointly or separately. Specifically, *unpartitioned* unit embeddings consist of a single vector that describe spectra, aperiodicity, $\log -F_0$ and voicing, while in *partitioned* unit embeddings consist of a super-vector of four vectors each individually representing spectra, aperiodicity, $\log -F_0$ and voicing. In both cases phone duration is embedded separately from the other streams. Figures 3 and 4 show the MOS-Naturalness and

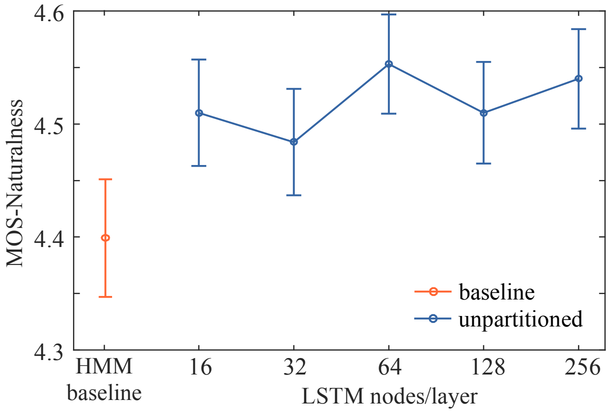

Figure 5: *Model size tuning for limited-domain TTS (Maps).*

confidence intervals of the two systems for several *target cost weights* varying from 0.5 to 2.0, as well as the baseline HMM-based system [2].

Limited-domain results in figure 3 show all proposed systems outperforming the baseline without statistical significance. However, given that all *unpartitioned* systems saturate around the maximum MOS level of 4.5 that raters assign to recorded speech, it is fair to claim that limited domain speech synthesis reached recording quality.

Open-domain results, in Figure 4, show that all proposed systems outperform the baseline; in most cases, substantially enough to be statistically significant without further AB testing. The best system, *unpartitioned* with a target cost weight of 1.5, outperforms the baseline by an impressive 0.20 MOS. The improvement is statistically significant since the confidence intervals do not intersect.

Further experiments omitted from this paper suggest that isometric training neither improves nor degrades MOS in our unit-selection framework. One possibility is that the distance preservation is approximate and not significant enough to alter the results meaningfully. Another possibility is that the effect is limited because we conduct the nearest neighbors search within each basetype.

The second experiment explores the relationship between MOS-Naturalness and model size. The best system from the previous experiment, *unpartitioned* with target cost weight of 1.50, is evaluated for LSTM layers with 16, 32, 64, 128, and 256 nodes per layer. A maximum size of 64 dimensions was used for each phone-embedding, while the (unit) diphone-embedding is constructed by concatenating two phone embeddings and further restricting the number of dimensions to 64 using Principal Component Analysis for computational reasons. Results for limited- and open-domain TTS are presented in figures 5 and 6 respectively. We observe that 64 LSTM nodes per layer is sufficient in terms of performance and quality. The confidence intervals in this test indicate that the proposed embeddings outperform the baseline with statistical significance, for open-domain as well as limited-domain TTS synthesis.

The third experiment compares our approach to WaveNet [5] in open-domain TTS (*WebAnswers*) using 1,000 randomly selected utterances from anonymized logs. These sentences are different from the ones used in the experiments above. The results are presented in figure 7, where we observe that the proposed method yields an statistically significant improvement of 0.16 MOS over the HMM-based baseline while it has a 0.13 MOS difference with the corresponding 24kHz WaveNet. The difference is much smaller when considering the much faster 16kHz WaveNet. Thus, the proposed method is in-between the baseline and the best reported TTS in terms of quality while it's computational load allows us to have it in production.

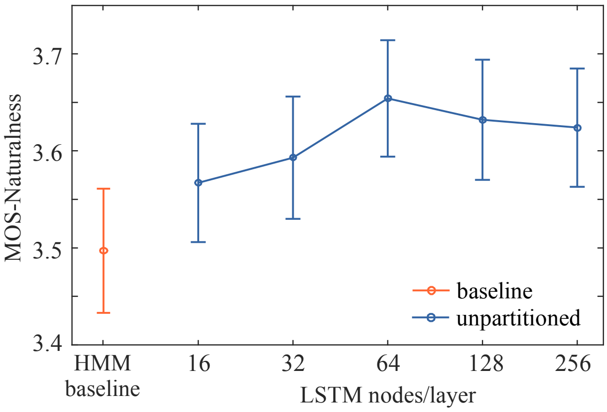

Figure 6: *Model size tuning for open-domain TTS (WebAnswers).*

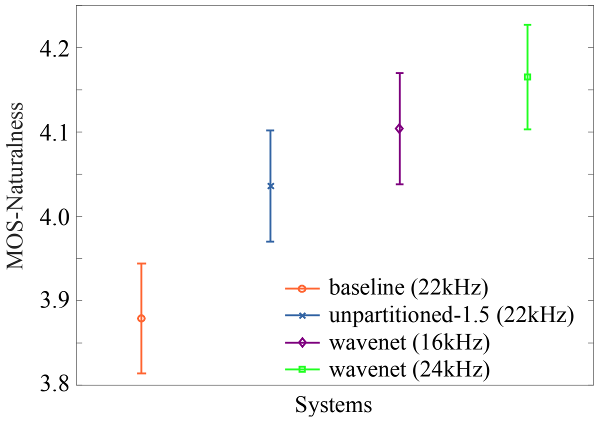

Figure 7: *Unit-selection vs. WaveNet for open-domain TTS (WebAnswers).*

## 5. Conclusion

We presented a novel LSTM-based sequence-to-sequence model for unit-selection that compresses the acoustics of variable-sized units to fixed-size embeddings and infers them from the linguistic features. Unit-selection is made by formulating the target cost as an $L_2$ distance in the embedding space. We report an improvement of up to 0.2 MOS for open-domain TTS synthesis, capping to the maximum of 4.5 MOS for limited-domain TTS synthesis and halving the distance between the previous HMM-based unit-selection solution [2] and WaveNet [5], while retaining low computational cost and latency.

This paper focussed on producing embeddings for a unit-selection system. However, it is also possible to produce speech from the embedding using the decoder network and a vocoder. Having a common embedding representation provides a simple framework for switching between unit-selection speech and SPSS at the unit-level.

# 6. References

[1] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proceedings of the Acoustics, Speech, and Signal Processing, 1996. On Conference Proceedings., 1996 IEEE International Conference - Volume 01*, ser. ICASSP '96. Washington, DC, USA: IEEE Computer Society, 1996, pp. 373–376. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.1996.541110

[2] X. Gonzalvo, S. Tazari, C. Chan, M. Becker, A. Gutkin, and H. Silen, "Recent advances in Google real-time HMM-driven unit selection synthesizer," in *Interspeech 2016*, 2016, pp. 2238–2242. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-264

[3] Y. Agiomyrgiannakis, "Vocaine the vocoder and applications in speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4230–4234.

[4] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak, "Fast, compact, and high quality LSTM-RNN based statistical parametric speech synthesizers for mobile devices," in *Interspeech 2016*, 2016, pp. 2273–2277. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-522

[5] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," in *Arxiv*, 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[6] J. Tao, Y. Zheng, Z. Wen, Y. Li, and B. Liu, "BLSTM guided unit selection synthesis system for Blizzard Challenge 2016," in *Proc. Blizzard Challenge*, 2016.

[7] T. Merritt, R. A. J. Clark, Z. Wu, J. Yamagishi, and S. King, "Deep neural network-guided unit selection synthesis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5145–5149.

[8] L.-H. Chen, Y. Jiang, M. Zhou, Z.-H. Ling, and L.-R. Dai, "The USTC system for Blizzard Challenge 2016," in *Proc. Blizzard Challenge*, 2016.

[9] X. Wang, S. Takaki, and J. Yamagishi, "An autoregressive recurrent mixture density network for parametric speech synthesis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017.

[10] Y. Saito, "Training algorithm to deceive anti-spoofing verification for DNN-based speech synthesis," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2017.

[11] X. Wang, S. Takaki, and J. Yamagishi, "Investigating very deep highway networks for parametric speech synthesis," in *Speech Synthesis Workshop 9*, 2016.

[12] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks." in *Interspeech*, 2014, pp. 1964–1968.

[13] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7962–7966.

[14] S. Achanta, K. R. Alluri, and S. V. Gangashetty, "Statistical parametric speech synthesis using bottleneck representation from sequence auto-encoder," 2016. [Online]. Available: http://arxiv.org/abs/1606.05844

[15] J. Weston, S. Bengio, and N. Usunier, "Large scale image annotation: learningtorank withjoint word-image embeddings," *Machine Learning*, vol. 81, no. 1, pp. 21–35, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10994-010-5198-3

[16] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov, "DeViSE: A deep visual-semantic embedding model," in *Advances in Neural Information Processing Systems 26*, 2013, pp. 2121–2129. [Online]. Available: http://papers.nips.cc/paper/5204-devise-a-deep-visual-semantic-embedding-model.pdf

[17] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4470–4474.