Polyglot speech generation with WaveNet

Csaba Zainkó, Bálint Pál Tóth, Géza Németh

Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics, Hungary
{zainko, toth.b, nemeth}@tmit.bme.hu

Abstract

WaveNet like deep learning architectures are capable of generating high quality speech while the control parameters of synthesis may be changed continuously. One of the possible parameters can be the language of the current sentence, phrase, word or phoneme. In this paper we show an experiment with WaveNet architecture that combines three languages (English, German and Hungarian) in one model. In our work a conditioned WaveNet was trained and tested with mono- and bilingual sentences.

This kind of polyglot speech generation is used e.g. in railway station announcement systems where the language may vary within the sentence.

Index Terms: WaveNet, polyglot, speech synthesis

1. Introduction

Polyglot speech synthesis refers to a computer system that can generate speech in a single voice for multiple languages. This is an extra feature compared to multilingual systems that use different voices for covering several languages.

Requirements for polyglot systems come from two main directions. Polyglot synthesis was introduced in Switzerland [1]. This was first implemented as a combination of three separate diphone-based systems derived from a speaker who spoke German, French and Italian at a native level. In this approach the aim is to generate messages in the three main Swiss official languages at the same system voice. The voice talent can speak all the languages and sufficient amount of recordings are available in all languages. In another approach there is no possibility to have recordings from the same voice for all the languages. In this case some sort of mapping has to be performed. Hidden Markov-model (HMM) based speaker adaptation was tested by phoneme mapping in Japan [2] and India [3]. The available quality is limited by the technology. A recent overview of polyglot synthesis can be found in [4].

Our application domain is public transport information systems, such as railway station announcements. In this case high voice quality is of utmost importance. For this reason, we have applied domain optimized corpus-based technology [5]. If the messages fall in the domain category the quality is very high in general. But there are some important exceptions as well. For example, train names change frequently, quite often with irregular pronunciations. In such cases new recordings may be necessary. In order to save this effort and reduce response time, alternative approaches have been considered. One possible solution may be using WaveNet [6]. In the initial phase the output of WaveNet could be used for words/sentences that cannot be covered well by the corpus-based system. In the long term if WaveNets can be operated real-time they can replace the corpus-based system.

Statistical parametric speech synthesis has been around for more than two decades, and it became a focused research area in speech sciences more than a decade ago. Before the rise of deep learning, hidden Markov-models were used to model speech parameters like spectral features, excitation and durations [7] and based on these parameters different vocoding techniques were used to generate synthetic speech [8]. With the arrival of high performance GPUs and novel results in neural networks, deep learning (DL) has become one of the most effective methods in machine learning. Modeling the parameters of the vocoder with deep learning for synthetic speech generation, including feed multi-layer perceptrons [8] and Long Short-Term Memory [10] has shown superior results to HMMs.

In September 2016 a novel approach, called WaveNet, that models raw audio with one dimensional dilated convolution was published [6]. Applying similar input features to the previous deep learning solutions (see Section 2.3. for details) besides raw audio Text-To-Speech (TTS) synthesis with high quality can be realized in WaveNet. In February 2017 an end-to-end speech synthesis system was proposed [11] that uses a similar approach to WaveNet for audio generation, completed with neural networks for grapheme-to-phoneme, segmentation, phone duration and fundamental frequency modeling.

2. System setup

2.1. Databases

The databases were designed for a corpus-based polyglot speech synthesis system [5]. The voice talent speaks native Hungarian and Romanian. The databases contain sentences in three languages: Hungarian, English and German.

Table 1: *Size of corpora.*

Language	**No. of sentences**	**Size**	**Duration**
Hungarian (original)	3261	655Mbyte	7.8 hours
Hungarian	693	202Mbyte	110 min
English	497	152Mbyte	88 min
German	671	163Mbyte	89 min

In this experiment, we decreased the size of the Hungarian part of the speech database to keep the balance between the languages. The original Hungarian corpus contains more than 3000 sentences, which is about 8 hours of speech. Because we have less than 2 hours from the other languages, we have selected a 110-minute-Hungarian part of the original corpus for training as described in Table 1. Both the English and the German corpus contain monolingual and bilingual sentences. In the bilingual sentences the second language is always Hungarian, and they are mainly railway station names.

The railway announcement system where this corpus is used focuses on local audience. In our region there are fewer native English or German passengers so the pronunciation of our voice talent fits this environment.

2.2. Model

We used the WaveNet architecture in our experiments, which is a generative model operating on raw audio. The idea of WaveNet is inspired by the PixelCNN architecture [12][13]. WaveNet utilizes stacked dilated causal convolutional (DCC) layers. The causal property ensures not to use any future information of the audio time-series, while dilatation helps to increase the receptive field and moderate computational costs. We used 40 stacked dilated causal convolutional layers, with (1,2,4,8,16,32,64,128,256,512)×4 dilatations. According to our preliminary experiments in case of 30 stacked layers the quality was worse, in case of 50 there wasn't any hearable difference in quality and resulted in slower training times. Deep Voice [11] also used 40 stacked dilated causal convolutional layers.

Every DCC layer has a gated activation function [13]. Both the depth of 1×1 convolutions in skip connections and the number of neurons in the output dense layer were 256. We conducted experiments with 512 and 1024 skip connections, however both increased training and inference times significantly. For filter depth we used 64, which was considered to result in better speech quality than 32 and 48. The architecture did not contain any dropout or pooling layers. The output was a softmax layer and the network was optimized to maximize the log-likelihood of training data.

2.3. Features

Based on the original WaveNet model we used 256 level μ-law quantization for the audio signal. Therefore, the output vector was 256 long one-hot encoded categorical vector. As inputs the audio signal and conditional features were used. From the audio signal the current and preceding, altogether $2^{10}+(2^{10}-1)\times3=4093$ (~256 ms) samples were used as inputs. The 256 long one-hot encoded vectors of the audio signal were mapped to 32 bit floats in case of the input.

For training and speech generation we calculated the following conditional input features for every millisecond. In case of features, that have a lower resolution (all of them except LogF0 and Voiced/Unvoiced flag) the values were held until the next segment. The input values were scaled to 0 mean and unit variance, except binary features like one-hot encodings and voiced/unvoiced flag.

- Currently, two preceding and two following phones (quinphones) are applied in one hot encoding. In case of two languages 86, in case of three languages 132 was the number of possible phones and they were represented in one-hot encoding. Thus 86×5=430 and 132×5=660 long sparse vectors were used for textual input, respectively.
- The type of prosody unit: defines if it is the first, center or last prosody unit, or there is only one prosody unit present. One-hot encoding was used, so this feature was represented by 4 inputs.
- The language (English, German, Hungarian) for each phone in one-hot encodings: 3 inputs.
- Segmental features:
    - Word number forward and backward: 2 inputs.
    - Phone number in word forward and backward: 2 inputs.
    - Percentile position within prosody unit and phone forward and backward: 4 inputs.
    - Phone duration in *ms*: 1 input.
- LogF0: 1 input.
- Voiced/Unvoiced flag: 1 input.

Altogether 448 and 678 features inputs for the two-language and the three-language systems, respectively.

2.4. Multilingual conditioning

The three languages are handled with one phone set which contains all possible phoneme codes. For two languages we used the English-Hungarian subsets, and for the third language we involved the German subset.

The training corpora contained words in several languages besides the three main ones. The pronunciation rule in the Hungarian railway announcement system is that the foreign target destinations are pronounced in Hungarian if there is a Hungarian equivalent and according to the language of the state (not in English or German, e.g. Venice is pronounced as Velence in Hungarian and Venezia in English and German announcements). It means that the system should have to handle about 20 languages, which would extremely increase the model size. To get rid of this effect, we simplify the transcription of these stations into Hungarian phoneme codes.

There are three codes to represent silence parts: one general silence, the other two representing the silence at the beginning and at the end of the sentences. The silence codes are language independent. The Hungarian code table contains 39 different phones while the English has 44 phones. The German subset involves 46 phones.

To handle the language changes the voice talent was trained to keep a short silence at the boundaries of the words where the language of the words changes. It means that neighboring phones don't interfere with each other, (at least) one language independent silence is inserted between them.

2.5. Training

The model is implemented in TensorFlow [14] and the training was performed on NVIDIA GPUs. We examined two different configurations. The first one contained only two languages: English and Hungarian. The other one was trained with three languages, English, German and Hungarian. The first (smaller) configuration uses less input features (448, details in Section 2.3). The three language version uses 678 features.

The waveforms were resampled to 16 kHz. The sentences were shuffled, the order of the sentences were language independently randomized. The batch size was 100.000 samples (6.25 sec raw audio). The sentences and feature vectors were concatenated if they were shorter than 6.25 sec utterances.

The two-language system was trained on an NVIDIA Titan X (Pascal), Cuda 8.0, CuDNN 5.1, TensorFlow 1.0.0, Ubuntu 14.04 configuration. The three-language version on an NVIDIA Titan X (Maxwell), Cuda 7.5, CuDNN 5.1, TensorFlow 0.9.0, Ubuntu 14.04 setup.

After each 500th epoch the models were saved and test sentences were generated on a different machine. At the two–language system one epoch lasted about 2.1 seconds, thus the whole training took more than one week (cca. 300.000 epochs). Because of the uncertainty of the architecture and slow generation times, stop criteria was not introduced. The three-language system is slower because of the higher number of features, therefore one epoch took about 2.4 secs. The latter system was trained until 200.000 epochs.

3. Speech generation

3.1. TTS systems

A TTS system contains several subsystems. The typical TTS chain is: text preprocessing, grapheme-to-phoneme conversion, prosodic prediction and waveform generation modules. WaveNet offers a solution only for waveform generation. In the current experiment we replaced the first modules of the TTS chain with derived data from natural speech. The advantage of this approach is that the quality of speech does not depend from other modules.

3.2. Wave generation

The speech-waveform generation is based on the Fast WaveNet algorithm [15]. It is faster than the native implementation of the WaveNet, because the redundant convolution operations were eliminated. The Fast WaveNet approach applies FIFO queues to keep the results of sub-calculations and use them later.

3.3. Polyglot textual input

We have used five different types of sentences as test stimuli. There are three monolingual and two bilingual groups. The monolingual groups contain Hungarian, English and German sentences. Most words of the sentences in the bilingual groups are English or German and they contain Hungarian units e.g. railway station or train names (see Fig.1).

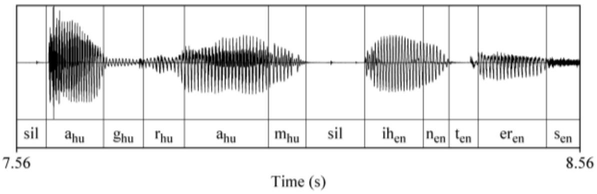

Figure 1: *Bilingual example …Agram Interc… (Agram: Hungarian train name, Intercity: English or German pronunciation)*

Because our research focused on wave generation, we used prosodic information of natural sentences as input of the WaveNet module. To compare the independent and dependent sentences of training data, we used some sentences from the training corpus, too.

3.4. Prosody modification

To test the flexibility of the model, a sub-experiment was performed, where we modified the speed of the input parameters of WaveNet. The speed was decreased and increased by 15, 30 and 50 % and a small speech expert group listened to the generated speech. We found, that the modification does not cause essential quality change.

3.5. Performance analysis

The generation is based on the Fast WaveNet architecture [15]. Because DCC layers force to calculate the waveform sample by sample, the GPUs' parallel computing capability cannot be heavily exploited. Memory operations between CPU and GPU and program control parts are more time-consuming than matrix calculus, thus the GPUs' main advantage is lost during inference. According to our experience using the CPU instead of the GPU results in better performance for speech generation. [11] reported similar results.

Table 2: *Generation speeds (Intel(R) Core(TM) i7-6850K CPU @ 3.60GHz, NVIDIA Titan X (Pascal).*

Method	Speed	Program lang.
GPU (Titan X)	50 samples/sec	Python 2.7
CPU (i7@ *3.60GHz)*	250 samples/sec	Python 2.7
CPU (i7@ *3.60GHz)*	710 samples/sec	C++

To reach real-time speech generation we need to generate 16000 samples/sec. The speed depends on the size and architecture of the model. E.g. the number of skip channels is an important factor, because it participates in several matrix operations.

To improve the speed of wave generation there are several different techniques. The structure of the model allows us to make some matrix operations parallel (e.g. for a few processor cores separate threads) or reuse memories. The choice of a proper programing language offers speed increase, e.g. C++ memory management ensures more control over the memory operations than basic Python. In this paper we have concentrated on the speech quality therefore we have used our C++ implementation which uses only one thread now. Multithreaded versions may offer faster generation. This solution is under implementation.

4. Evaluation

To evaluate the test samples we conducted a web-based MUSHRA (MUltiStimulus test with Hidden Reference and Anchor) listening test [16]. The advantage of the MUSHRA test is that it allows evaluating multiple waveform samples in a single trial without breaking the task. It is faster than pair comparison tests.

4.1. Test setup

In the test there were 8 different sentences with 9 versions of the system. The original reference sentence was also evaluated. The order of test sets and the systems were set randomly. Different subjects listened to the samples in a different order.

Some of the test samples were generated by two-language systems (b, c, d), the others were generated by the three-language systems (e, f, g, h). There were two deliberately bad systems (i, j), which are necessary in MUSHRA tests as anchors (see details in Table 3).

Table 3: *Main parameters of the systems.*

System	b	c	d	e	f	g	h	i	j
No. lang.	2	2	2	3	3	3	3	3	3
Epochs	160k	228k	252k	114k	158k	175k	178k	67k	78k

The test contains 4 monolingual and 4 bilingual sentences. The length of sentences was adjusted to typical railway station scenarios, so the average sentence length was quite long: 77 phonemes (9.7 sec). The shortest one contained 22 phonemes (2.9 sec), the longest sentence included 117 phonemes (19.7 sec).

The subjects had to move a horizontal slider between 1-100 (1 worst, 100 best). The following headings were uniformly distributed over the slider: Highly unnatural, Unnatural, Intermediate, Natural, Highly natural.

4.2. Test results

Altogether 12 (4 female and 8 male) listeners participated in the web based evaluation. The average age of the subjects was 37 years (ranging from 15 to 70). Most participants were native Hungarian speakers. There is no known hearing impairment of the subjects.

Table 4: *Average values of answers.*

Samples	a	b	c	d	e	f	g	h	i	j
1 (Hu)	91	68	55	53	49	69	54	44	31	29
2 (En-Hu)	92	44	55	55	53	51	46	56	20	27
3 (En)	91	35	42	37	26	41	23	37	23	33
4 (De-Hu)	92				44	49	55	22	30	23
5 (De)	84				35	57	48	56	20	29
6 (En-Hu)	95	49	56	47	41	35	30	33	4	23
7 (En-Hu)	95	33	44	25	34	37	37	41	8	10
8 (En-Hu)	91	35	44	24	38	44	47	44	11	27
Average	91	44	49	40	40	48	42	42	18	25

The average values of answers are shown in Table 4. Column 'a' shows the reference speech values. The fourth and fifth sample sentences are not generated with system b, c, d because they are only two-language (En-Hu) systems.

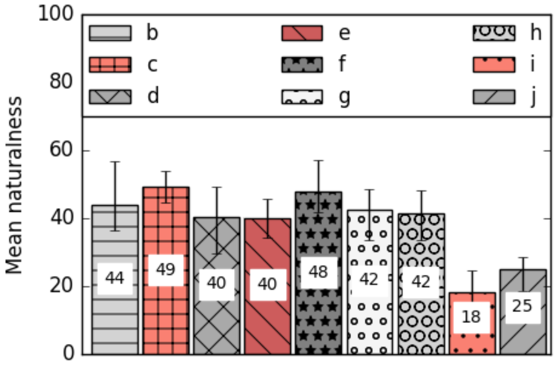

Figure 2: *Overall results of the systems.*

The last two systems were dedicated as bad systems, the results confirm that. Overall averages are presented in Fig.2. The best system was system c, which is a two-language-system after 228 000 epochs. Significant difference was found only between system c and d (Mann-Whitney-Wilcoxon ranksum test, p<0.05) among the two-language variants.

The most natural three-language-system was system f, which was saved after 158 000 epochs. System f is significantly better than the other three-language-systems (p<0.05).

Table 5: *Details of system f.*

Sentence language	EN	DE	HU	EN-HU	DE-HU
Scores	41	57	69	42	49

According to the results of system f (Table 5), we found, that the Hungarian monolingual and the German-Hungarian bilingual samples were the best and the English samples were the worst.

5. Conclusions

In this study we have examined the possibilities of applying the WaveNet technology in a polyglot speech synthesis context. It can be concluded that it was possible to create a single WaveNet model for the three languages. Taking into account the relatively small training data (less than 2 hours / language) it is encouraging that the best systems achieved intermediate average quality in the MUSHRA test with quite long (up to nearly 20 seconds) mixed language sentences. The Hungarian monolingual and the German-Hungarian bilingual samples produced the most natural outputs. It may be due to the similar phonetic structure of these languages. The generalization capabilities of the system seem to be also promising. It is a further advantage that it is not necessary to adapt to a vocoder technique as in the case of traditional HMM and DNN approaches.

In the future we intend to test other input feature possibilities (e.g. speaking rate). In order to create a full TTS system, timing and F0 prediction has to be adapted [17][18][19][20]. A promising further research direction is the creation of multi-speaker and multi-lingual models. The training and generation speeds should also be improved.

6. Acknowledgements

This research was partly supported by the VUK (AAL-2014-1-183) and the EUREKA / DANSPLAT projects. We would like to thank Tamás Gábor Csapó for preparing the web-based test and thank the subjects for participating in the listening test. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

7. References

[1] C. Traber, B. Pfister, "From multilingual to polyglot speech synthesis". Proceedings of Eurospeech'99, Budapest, Hungary, pp. 835–838.

[2] J. Lattore, K. Iwano, and S. Furui, "New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer," Speech Communication, vol. 48, pp. 1227-1242, 2006.

[3] B. Ramani, M.P. Jeeva, P. Vijayalakshmi, T. Nagarajan, "Voice conversion-based multilingual to polyglot speech synthesizer for Indian languages", Proceedings of TENCON 2013, 22-25 Oct. 2013, Xi'an, China pp. 1 - 4.

[4] Bidisha Sharma & S. R. Mahadeva Prasann, "Polyglot Speech Synthesis: A Review", IETE Technical Review, pp. 1-24, 2016/08/02

[5] Csaba Zainkó, Mátyás Bartalis, Géza Németh, Gábor Olaszy, "A Polyglot Domain Optimised Text-To-Speech System for Railway Station Announcements" Proc. of INTERSPEECH 2015. pp. 1236-1240.

[6] Oord AV, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K. "WaveNet: A

generative model for raw audio." arXiv preprint arXiv:1609.03499. (2016 Sep 12.)

[7] Zen, Heiga, Keiichi Tokuda, and Alan W. Black. "Statistical parametric speech synthesis." Speech Communication 51, no. 11 (2009): pp. 1039-1064.

[8] Tamás Gábor Csapó, Géza Németh, "Modeling irregular voice in statistical parametric speech synthesis with residual codebook based excitation", In: IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING, vol. 8, no. 2, 2014, pp. 209-220.

[9] Ze, H., Senior, A. and Schuster, M., 2013, May. "Statistical parametric speech synthesis using deep neural networks". In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013, pp. 7962-7966.

[10] Fan, Yuchen, Yao Qian, Feng-Long Xie, and Frank K. Soong. "TTS synthesis with bidirectional LSTM based recurrent neural networks." In Interspeech, 2014 pp. 1964-1968.

[11] Arik, S. O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., Sengupta, S. & Shoeybi, M. (2017). "Deep Voice: Real-time Neural Text-to-Speech." arXiv preprint arXiv:1702.07825.

[12] Oord, Aaron van den, Nal Kalchbrenner, and Koray Kavukcuoglu. "Pixel recurrent neural networks." arXiv preprint arXiv:1601.06759 (2016).

[13] van den Oord, Aaron, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, and Alex Graves. "Conditional image generation with PixelCNN decoders." In Advances in Neural Information Processing Systems, pp. 4790-4798. 2016.

[14] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015. Software available from tensorflow.org.

[15] Paine, T. L., Khorrami, P., Chang, S., Zhang, Y., Ramachandran, P., Hasegawa-Johnson, M. A., & Huang, T. S. (2016). "Fast Wavenet Generation Algorithm" arXiv preprint arXiv:1611.09482

[16] ITU-R Recommendation BS.1534: "Method for the subjective assessment of intermediate audio quality," 2001.

[17] Bálint Pál Tóth, Balázs Szórádi, Géza Németh, "Improvements to Prosodic Variation in Long Short-Term Memory Based Intonation Models Using Random Forest", In: 18th International Conference on Speech and Computer SPECOM 2016, Budapest, Magyarország, 2016, p. 9

[18] Bálint Pál Tóth, Kornél István Kis, György Szaszák, Géza Németh, "Ensemble Deep Neural Network Based Waveform-Driven Stress Model for Speech Synthesis", In: 18th International Conference on Speech and Computer SPECOM 2016, Budapest, Magyarország, 2016, p. 8

[19] Péter Nagy, Géza Németh, "DNN-Based Duration Modeling for Synthesizing Short Sentences", In: Speech and Computer, Budapest, Magyarország, 2016, pp. 254-261

[20] Bálint Pál Tóth, Tamás Gábor Csapó, "Continuous Fundamental Frequency Prediction with Deep Neural Networks", In: 2016 European Signal Processing Conference (EUSIPCO 2016), Budapest, Magyarország, 2016, pp. 1348-1352