Towards End-to-end Text to Speech Synthesis in Macedonian Language

Marija Neceva, Emilija Stoilkovska, Hristijan Gjoreski
mneceva@gmail.com, emi.stoilkovska@gmail.com, hristijang@feit.ukim.edu.mk
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
Skopje, N. Macedonia

ABSTRACT

A text-to-speech (TTS) synthesis system typically consists of multiple stages: text analysis frontend, an acoustic model and an audio synthesis module. Building these components often requires extensive domain expertise and may contain brittle design choices. The paper presents an end-to-end deep learning approach to speech synthesis in Macedonian language. The developed model uses the Google's Tacotron architecture and is able to generate speech out of text from multiple speakers using attention mechanism. It consists of three parts: an encoder, an attention-based decoder and a post-processing network. The model was trained on a dataset recorded by five, mixed gender speakers, resulting in 25.5 hours of data, or 13,101 pairs of text-speech segments. The results show that the model successfully generates speech from text data, which was empirically shown using a quantitative questionnaire answered by 42 subjects.

KEYWORDS

text-to-speech, deep learning, tacotron, multi-speaker, seq2seq, text, audio, attention

1 INTRODUCTION

Modern TTS pipelines are complex [1]. For example, statistical parametric ones have a text frontend, extracting various linguistic features, a duration model, an acoustic feature prediction model and a complex signal-processing-based vocoder [2][3]. These components usually require extensive domain expertise, are laborious to design and must be trained independently. Consequently, errors from each component may compound. Otherwise, implementing an integrated end-to-end TTS system offers many advantages. First, it can be trained on <text, audio> pairs with minimal human annotation. It also alleviates the need for laborious feature engineering. Further, it allows rich conditioning on various attributes, such as speaker or language, or high-level features like sentiment. Similarly, adaptation to new data might also be easier. Finally, a single model is likely to be more robust than a multi-stage. All these advantages imply that an end-to-end system allows training on huge amounts real world data. But knowing that TTS is a large-scale inverse problem and due to existence of different pronunciations or speaking styles, decompressing a highly compressed source text into audio may cause difficulties in the learning task of an end-to-end model. The main problem is coping with large variations at the signal level for a given input. Moreover, unlike end-to-end speech recognition [4] or machine translation [5], TTS outputs are continuous, and much longer than input sequences. Mainly referring to the advantages of end-to-end systems, this paper proposes an implementation of Google's Tacotron model as a TTS system for Macedonian language. Tacotron is an end-to-end generative TTS model based on the sequence-to-sequence model (seq2seq) [6] with attention paradigm [7]. This model takes characters as input and outputs raw spectrogram. We implemented our own version of Tacotron, based on few published articles. What we kept is their deep learning architecture, but made some changes in model's hyper parameters and other utilities (like known symbols, numbers etc.). That way the model was adapted to work with Cyrillic. Given <text, audio> pairs, our Tacotron model was trained completely from scratch only on our dataset. It does not require phoneme-level alignment, so it can easily scale to using large amounts of acoustic data with transcripts.

2 RELATED WORK

WaveNet [8] is a powerful, non end-to-end, generative audio model which works well for TTS synthesis. It is used as a replacement of the vocoder and acoustic model of the system. It can be slow due to its sample-level autoregressive nature. It also requires conditioning on linguistic features from an existing TTS frontend.

Deep Voice [9] is a neural model which replaces every component in a typical TTS pipeline by a corresponding neural network. However, each component is independently trained, and it's nontrivial to change the system to train in an end-to-end fashion.

Wang et. al [10] presents one of the first studies of end-to-end TTS using seq2seq with attention. However, it requires a pre-trained hidden Markov model (HMM) aligner to help the seq2seq model learn the alignment and a vocoder due to predicting vocoder parameters. Furthermore, the model is trained on phoneme inputs with possibilities of hurting the prosody and producing limited experimental results.

Char2Wav [11] is an independently developed end-to-end model that can be trained on characters. However, it still predicts vocoder parameters before using a SampleRNN neural vocoder [12] and their seq2seq and SampleRNN models need to be separately pre-trained.

MAIKA is a Macedonian TTS project that was made public few months ago. However, there is no documentation of how it works. Therefore, it is technically challenging to compare with a system that only has web interface which generates sound.

eSpeak is an open source TTS project that also supports Macedonian language. The documentation states that the Macedonian model is based on the Croatian - which has its limitations since the Macedonian language is quite different, especially the pronunciation and the grammar.

3 MODEL ARCHITECTURE

The backbone of Tacotron is a seq2seq model with attention [7][13]. Figure 1 illustrates the model, which includes an encoder, an attention-based decoder, and a post-processing net. At a high-level, this model takes characters as input and produces spectrogram frames, which are later converted to waveforms. These components are described below.

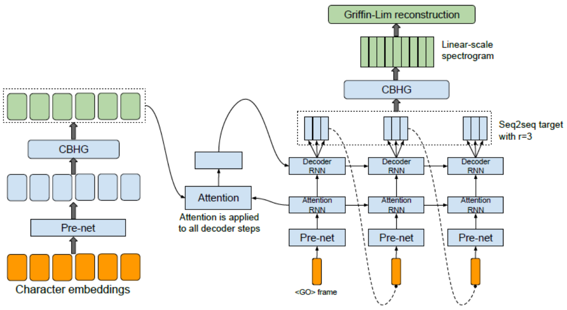

Figure 1: Model architecture

3.1 CBHG Module

CBHG is a module for extracting representations from sequences. It consists of bank of 1-D convolutional filters, followed by highway networks [14] and a bidirectional gated recurrent unit (GRU) [15]. The input sequence is first convolved with k sets of 1-D convolutional filters. These filters explicitly model local and contextual information (creating unigrams, bigrams, up to k-grams). Next the convolution outputs are stacked together and max pooled along time to increase local invariances. Further the processed sequence is passed to a few fixed-width 1-D convolutions, whose outputs are added with the original input sequence via residual connections [16]. Batch normalization [17] is used for all convolutional layers. Moreover, the fixed-width convolution outputs are fed into a multi-layer highway network to extract high-level features. Finally, a bidirectional GRU RNN has been stacked on top, extracting sequential features from both forward and backward context.

3.2 Encoder

The encoder extracts robust sequential representations of text. The input to the encoder is a character sequence, with each character represented as a one-hot vector and embedded into a continuous vector. Onto each embedding is applied a set of non-linear transformations, known as "pre-net". The "pre-net" is represented as a bottleneck layer with dropout, helping convergence and improving generalization. A CBHG module transforms the "pre-net" outputs into the final encoder representation used by the attention module. Moreover, CBHG-based encoder reduces overfitting and makes fewer mispronunciations than a standard multi-layer RNN encoder.

3.3 Decoder

Tacotron model uses a content-based *tanh* attention decoder [18], where a stateful recurrent layer produces the attention query at each decoder time step. The input of decoder's RNN is formed by concatenating the context vector and the attention RNN cell output. Decoder's internal structure is a stack of GRUs with vertical residual connections [5], used for speeding up convergence. A simple fully-connected output layer is used to predict the decoder targets. Its target is 80-band mel-scale spectrogram, later converted to waveform by a post-processing network. It predicts multiple, non-overlapping, output frames at each decoder step. Predicting r frames at once divides the total number of decoder steps by r, which reduces model size, training and inference time and increases convergence speed. This is likely because neighboring speech frames are correlated and each character usually corresponds to multiple frames, plus emitting multiple frames allows the attention to move forward early in training. For defining the input of the next decoding step "teacher forcing" mechanism is used, pointing that on each time step, decoder's input is the ground-truth value of the previous predicted decoder output.

3.4 Attention mechanism

Attention mechanism is applied in order to "learn" mappings between input and output sequences through gradient descent and back-propagation. It is used as a way for the decoder to learn at which time step, which internal state of the encoder deserves more attention when generating its current output. The whole process of calculating the attention weights and using them to form the decoder input has been illustrated in Figure 2.

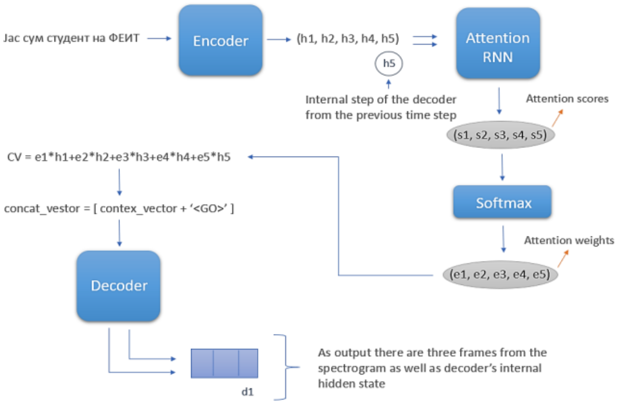
Јас сум студент на ФЕИТ
Encoder
(h1, h2, h3, h4, h5)
h5
Attention RNN
Internal step of the decoder from the previous time step
Attention scores
(s1, s2, s3, s4, s5)
CV = e1*h1+e2*h2+e3*h3+e4*h4+e5*h5
Softmax
concat_vestor = [contex_vector + '<GO>']
Attention weights
(e1, e2, e3, e4, e5)
Decoder
d1
As output there are three frames from the spectrogram as well as decoder's internal hidden state

Figure 2: What is behind the attention mechanism

3.5 Post-processing net and waveform synthesis

The post-processing net is converting the seq2seq target to a form that can be synthesized into waveforms [20][21]. Since Griffin-Lim has been used as a synthesizer, the post-processing net learns to predict spectral magnitude, sampled on a linear-frequency scale. The Griffin – Lim algorithm allows convergence towards estimated phase layer. Phase's quality depends on the number of iterations applied. Although more iterations may lead to overfitting, better audio is produced. Within our setup, Griffin-Lim converges after 50 iterations even though 30 iterations seems to be enough.

3.6 Model parameters

The log magnitude spectrogram is obtained using Hann windowing with 50 ms frame length, 12.5 ms frame shift, and 2048-point FT. 24 kHz sampling rate has been used for all experiments. For both seq2seq decoder (mel-scale spectrogram) and post-processing net (linear-scale spectrogram) a simple L1 loss with equal weight has been used. The model has been trained using a batch size of 4, where all sequences are padded to a max length.

4 DATASET

There is no public dataset of audio data in Macedonian language, therefore we had to create one. We used publicly available books in Macedonian from the website of the National Association of the Blind of the Republic of North Macedonia. The books have been recorded by 5 speakers, 3 male and 2 female. They are segmented using an algorithm which separates input audio based on silence length and threshold. Silence length varies between 700 – 1000 ms. The audio clips were additionally padded with 700 ms at both beginning and end to avoid sudden cut offs.

Next, the audio files were transcribed manually, aided by the written version of the audio book. The transcriptions are void of any punctuation, capitalization, or any special characters, including numbers. They include only the 31 letters from the Macedonian alphabet and the space character to separate between words. The reason for this is that the initial dataset was also used for another task (Speech Recognition) and the researchers removed the punctuations. In this phase we could not retrieve the original raw data that includes the punctuation. The final dataset contains 13,101 audio files and transcripts in Macedonian language [25]. Additional statistics about the dataset are listed in Table 1.

To be mentioned, the goal of the dataset is not the dataset itself, but how we can develop a deep learning, end to end, multi-speaker TTS for Macedonian language. Detailed language analysis of the dataset is planned for another study, in which the focus will be more on the linguistically part of the dataset.

Table 1: Dataset statistics

Total Clips	13 101
Total Words	188 521
Distinct Words	28 791
Total duration	25:36:20
Mean Clip Duration	7.04 sec
Min Clip Duration	0.73 sec
Max Clip Duration	97.6 sec (1.37 min)

5 TRAINING AND EVALUATION

5.1 Training

During the training phase there is an output produced on every 1000th step. It takes few seconds for an output to be produced. Each output contains five files, three of which give information about the model formed up to that step, while the other two are an alignment plot and an audio file synthesized by that mode. The synthesized audio file is used for checking the quality of the current model. The alignment plot shows if the decoder has learned which input state of the encoder is important for producing its current output. That means if there is an "A" on input, "A" should be produced as sound for output. As a good alignment plot is considered the one who looks like a diagonal line. This system was trained for 5 days, reached 412 000 steps and got 412 different models. The system started showing a good alignment on 63 000th step. The last model was chosen as referent one. Its training and test results sound much better and were more understandable than those generated from the other models.

5.2 Evaluation

To estimate the model's performance, we used 10, out of 14 random sentences as test examples. The results show that more than half of the synthesized audio files [22] were successfully representing the input sequence of the model. This was empirically shown using a quantitative questionnaire [23] answered by 42 subjects, 10 IT experts and 32 general public volunteers. The questionnaire was made up of 10 stages, for each of the 10 audio files. The reason for choosing 10 test examples was to make the questionnaire more compact, smaller and quicker for the evaluators. Each stage contains 3 sub questions for the currently observed audio file. The Mean Opinion Score (MOS) [24] was used as a measure for answering i.e. scoring each one of it. MOS is a measure of audio quality. It is a subjective measurement used to test the listener's perception of the audio quality and clarity. A group of 42 subjects were asked to do the questionnaire. Each audio file required to be scored with a score from 1-5 in terms of three criterions: naturalness, intelligibility and accuracy. Where naturalness stands for the similarity of produced audio file with the natural human speech, intelligibility or clarity of spoken words and accuracy or how much the spoken sequence corresponds with the original, required to be spoken text.

The results from the questionnaire are shown in Table 2. Each row of the table represents the MOS for one of the three criterions, calculated separately for experts and volunteers. The calculations are done by summing the scores for each criterion and consequently averaging it. By analyzing the results for each criterion is clear that, the experts score the model's performance better compared to the volunteers. Looking at the total score, experts evaluated the model's performance for 0.265 better than the volunteers. We speculate that the reason for this might be that when the experts are evaluating the model they also take into account the technical challenges and aspects of such system. On the other hand the volunteers simply evaluate the sound and its quality.

Additionally, in Figure 3 and Figure 4 we show the box-plots for the answers given by the experts and the volunteers respectively. The figures show that the accuracy is the characteristic that achieves the highest score, and the naturalness is the characteristic that achieved the lowest score. We speculate that the reason for low naturalness score is the presence of sudden pauses when words should be spoken or existence of mumbling instead of clear pronunciation. There are only few such occurrences.

Table 2: MOS Score results

	MOS Score	
	Experts	Volunteers
Accuracy	4.8	4.6
Intelligibility	4.5	4.2
Naturalness	4.1	3.9
Total	4.5	4.2

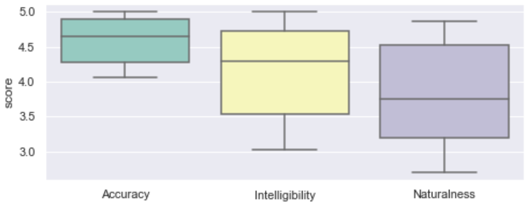

Figure 3: Box plot of all grades given by the volunteers

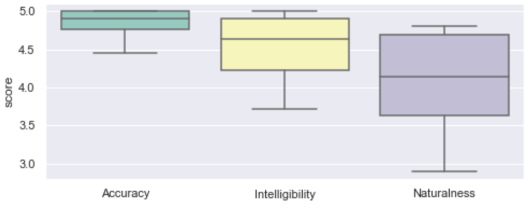

Figure 4: Box plot of all grades given by the IT experts

6 CONCLUSION

The paper presented an end-to-end deep learning approach to speech synthesis in Macedonian language. The developed model uses the Google's Tacotron architecture and generates speech out of text from multiple speakers using attention mechanism. The approach consists of three parts: an encoder, an attention-based decoder and a post-processing network. The model was trained on a dataset recorded by five, mixed gender speakers, resulting in nearly 25.5 hours of data. The results show that the model successfully generates speech from text, which was empirically shown using a quantitative questionnaire answered by 42 subjects.

To the best of our knowledge, this is the first end-to-end multi-speaker deep learning model for Macedonian language. We strongly believe that this will be a benchmark and motivation for future studies and finally to have a decent TTS system for Macedonian - which has significant societal impact.

Some of the limitations of the model are the gender diversity of speakers and the limited dataset. There is definitely room for improvement, and probably the dataset plays a crucial role in it. However, the data collection process is extensive and very time consuming task. With the given dataset we cannot estimate or empirically evaluate how much more data is needed to achieve state-of-the-art intelligibility and naturalness of artificially created speech. Additionally, in a few of the generated samples there are pauses at places where a word should be spoken. The reason for this is when the model generates sound, it uses character embeddings with specific ordering, learned during training. If those embeddings have never been seen during training, the model will not be able to properly pronounce them. Note that this is not the case with all of the words not being present in the training data, but in very rare occasions. Normally, the model will still generate speech even though a word is not present in the dataset.

ACKNOWLEDGEMENT

We are thankful for the support of the NVIDIA Corporation and their generous donation of a Titan XP GPU.

REFERENCES

[1] P.Taylor. Text-to-speech synthesis. Cambridge university press, 2009.
[2] H. Zen, K.Tokuda,A.W.Black. Statistical parametric speech synthesis. Speech Communication, 51(11):1039–1064, 2009.
[3] Y.Agiomyrgiannakis. Vocaine the vocoder and applications in speech synthesis. In Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on, pp. 4230–4234. IEEE, 2015.
[4] W.Chan, N.Jaitly, Q.Le, and O.Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, pp. 4960– 4964. IEEE, 2016.
[5] Y.Wu, M.Schuster, Z.Chen, Q.V.Le, M.Norouzi,W.Macherey, M.Krikun, Y.Cao, Q.Gao, K.Macherey. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.
[6] I.Sutskever, O.Vinyals,Q.V.Le. Sequence to sequence learning with neural networks. In Advances in neural information processing systems, pp. 3104–3112, 2014.
[7] D.Bahdanau, K.Cho, Y.Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.
[8] A.Oord, S.Dieleman, H.Zen, K.Simonyan, O.Vinyals, A.Graves, N.Kalchbrenner, A.Senior, K.Kavukcuoglu. WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.
[9] S.Arik, M.Chrzanowski, A.Coates, G.Diamos, A.Gibiansky, Y.Kang, X.Li, J.Miller, J.Raiman, S.,M.Shoeybi. Deep voice: Realtime neural text-to-speech. arXiv preprint arXiv:1702.07825, 2017.
[10] W.Wang, S.Xu, B.Xu. First step towards end-to-end parametric TTS synthesis: Generating spectral parameters with neural attention. In Proceedings Interspeech, pp. 2243–2247, 2016.
[11] J.Sotelo, S.Mehri, K.Kumar, J.F.Santos, K.Kastner, A.Courville, Y.Bengio. Char2Wav: End-to-end speech synthesis. In ICLR2017 workshop submission, 2017.
[12] S.Mehri, K.Kumar, I.Gulrajani, R.Kumar, S.Jain, J.Sotelo, A.Courville, Y.Bengio. SampleRNN: An unconditional end-to-end neural audio generation model. arXiv preprint:1612.07837, 2016.
[13] O.Vinyals, Ł.Kaiser, T.Koo, S.Petrov, I.Sutskever, G.Hinton. Grammar as a foreign language. In Advances in Neural Information Processing Systems, pp. 2773–2781, 2015.
[14] R.K.Srivastava, K.Greff, J. Schmidhuber. Highway networks. (2015).
[15] J.Chung, C.Gulcehre, K.H.Cho, Y.Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555, 2014.
[16] K.He, X.Zhang, S.Ren, J.Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.770–778, 2016.
[17] S.Ioffe, C.Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.
[18] O.Vinyals, Ł.Kaiser, T.Koo, S.Petrov, I.Sutskever, G.Hinton. Grammar as a foreign language. In Advances in Neural Information Processing Systems, pp. 2773–2781, 2015.
[19] D.Kingma ,J.Ba. Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015.
[20] Y.Masuyama, K.Yatabe, Y.Koizumi, Y.Oikawa, N.Harda (2019): Deep Griffin – Lim Iteration.
[21] J.Wodecki (2018): Intuitive explanation of the Griffin – Lim algorithm.
[22] Synthesized test audio files: https://drive.google.com/drive/folders/1LkgKAKcD9qNMw_3stbHEhszxhrPyPmAA?usp=sharing.
[23] Quantitative questionnaire used for evaluation of the model: https://docs.google.com/forms/d/e/1FAIpQLSeJJJVRjU3tzbLi1mix9buNOs002GFaTvSp9TVO752OCPNUvA/viewform?fbclid=IwAR1bLE8hrEALj7MwHkAgDKrf0JfyClD-DTuCiGdJ8Nc68Jl1XYv_1_MRxoE.
[24] P.C. Loizou. Speech Quality Assessment. University of Texas-Dallas, Department of Electrical Engineering, Richardson, TX, USA.
[25] M.Trajanoska, H.Gjoreski. Towards end-to-end Speech Recognition in Macedonian Language. BalkanCom (2019).