

Improving Naturalness of Neural-based TTS system Trained with Limited Data

Layan Sawalha

Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, Hungary

Layan.sawalha@edu.bme.hu

Mohammad Salah Al-Radhi

Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics

Budapest, Hungary

malradhi@tmit.bme.hu

Abstract— In this paper, we examined different approaches, including a neural network speech synthesis system and a non-autoregressive text-to-speech (TTS) model. We demonstrated how a baseline system based on Merlin can be used for TTS synthesis to produce a voice that is similar to a human. Typically, this system is only implemented with a front-end text processor and a WORLD vocoder. Here, we adapted Continuous and Ahocoder vocoders, and then we evaluated the effectiveness of each vocoder's techniques in producing high-quality speech. In the non-autoregressive TTS model, we implemented the state-of-the-art FastSpeech2 system, which provided high-quality speech synthesis in a timely manner without controllability and robustness issues. We focused on integrating a different language with limited data while maintaining high-quality speech synthesis. Through objective and subjective evaluations, we verified that our method could outperform the baseline system with full data.

Keywords—TTS, Neural Network, Deep Learning.

I. INTRODUCTION

In our daily lives, communication is needed and used in every aspect of our lives, each person's unique voice remains one of the main characteristics of human speech. It's an effective way of identifying a person, even though there are many alternatives for verbal communication, we cannot deny that it can never replace it. Speech processing can be used in various applications, such as single-channel enhancement, emotional conversion, bandwidth extensions of narrowband speech, and voice conversion [1]. Text-to-speech or TTS is a software that reads text and converts it into speech. TTS converts any text-based message into a verbal message. TTS is an evolving field that provides faster messages with consistency, time, and money saving. You can prepare your message in text and send it as a voice, so you don't have to record yourself, you can also make it consistent and professional by making the communications all by the same voice. TTS is beneficial to business applications by assisting them in delivering a variety of notifications simultaneously [2]. Here, various technologies are discussed, highlighting their main specifications, differences, and methodologies. A neural network speech synthesis Merlin [3] is implemented with three different vocoders to find the best voice quality.

A. Problem Definition

Text-to-Speech is an evolving field, where different systems have been developed for a long period and may be

used for various purposes. One of the main challenges with synthesized voice is that it often sounds robotic, inexpressive, and not authentic. To overcome this, we aimed to integrate, evaluate, and implement different vocoders to find the highest quality of synthesized speech that met the requirements for having a genuine human voice in a robot device. TTS is often implemented or applicable in one language, usually English, due to the availability of good infrastructure, such as datasets, and lower complexity compared to other languages. In this study, we integrated the Arabic Language, which was very challenging as there are very few available free-speech corpora. Our ultimate goal is to be able to customize TTS to not only different languages but also to different personal voices with limited data, as it can be costly to collect a sufficient amount of dataset from the target voice.

In order to achieve high Text-to-Speech results, a large dataset is often required, which can be a barrier when implementing it in different languages. Nowadays, there are almost seven thousand spoken languages with insufficient datasets, which constrains the applicability of TTS.

II. METHODOLOGY

A. Text-to-Speech with Full Data

In this work, the Merlin toolkit¹ was implemented, using full data which refers to the complete dataset. Merlin has some of the features required to build a text-to-speech system. It necessitates the use of a front-end and a vocoder. Merlin is implemented using only the WORLD vocoder [4] but in this study, we integrated different vocoders (i.e., Continuous [5] and Ahocoder [6] vocoders.

The vocoder is a component of various speech synthesis applications such as TTS, voice conversion, etc. There are different types of vocoders with similar strategies [7], the first stage is the analysis which is used to convert speech into parameters that present the vocal fold signal and vocal tract filter separately into the excitation signal. In the synthesis stage, the parameter is used to reconstruct the original speech signal.

The baseline WORLD vocoder was used in the first part of the experiment which is open-source speech analysis, modification, and synthesis. It can calculate the fundamental frequency (F0), aperiodicity, and spectral envelope, as well as create speech using only estimated parameters [8]. The WORLD vocoder is based on source-filter separation, i.e.

¹ <https://github.com/CSTR-Edinburgh/merlin>

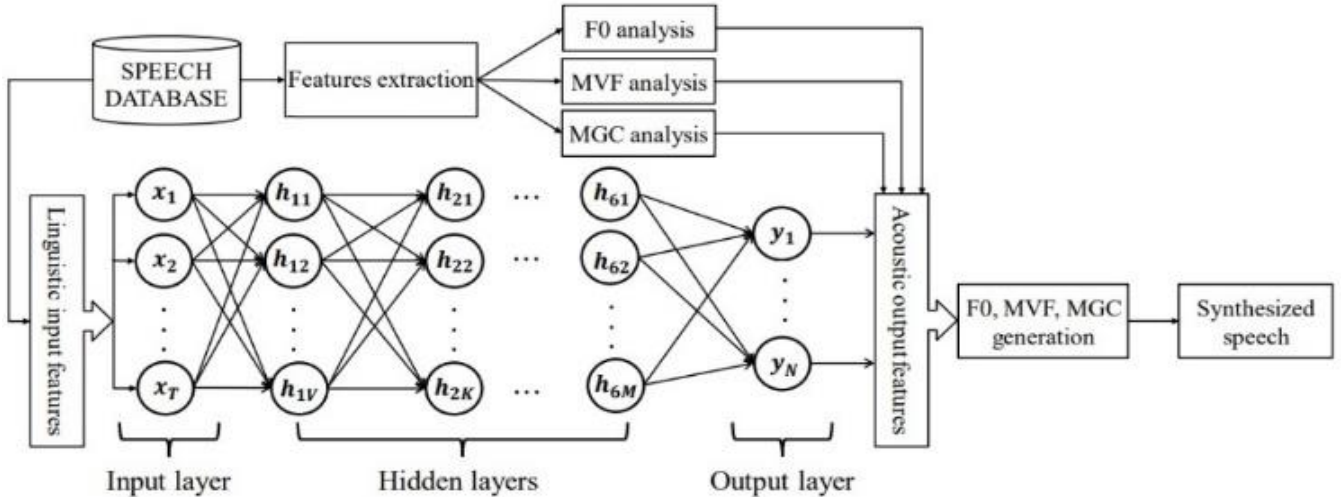


Figure 1: Schematic diagram of the proposed TTS system method based on Continuous and Ahocoder vocoders.

models separately the excitation (with F0 and aperiodicity) and the spectral envelope.

The second vocoder we implemented was the Continuous vocoder which is used to overcome the shortcoming of discontinuity in the speech parameters and the computational complexity of modern vocoders. The most important thing about this vocoder is that it does not need to have voiced/unvoiced decisions. During the analysis phase, F0 is calculated on the input waveforms of a simple continuous pitch tracker [9]. In regions of creaky voice and in case of unvoiced sounds or silences, this pitch tracker interpolates F0 based on a linear dynamic system and Kalman smoothing. After this step, Maximum Voiced Frequency (MVF) is calculated from the speech signal, resulting in the MVF parameter [10]. In the next step, 24-order MelGeneralized Cepstral analysis (MGC) is performed on the speech signal with $\alpha=0.42$ and $\gamma=-1/3$ [11]. In all steps, a 5 ms frameshift is used. The results are the F0, MVF, and MGC parameter streams.

The last vocoder we integrated into the Merlin framework is Ahocoder, which divides voice frames into three streams: F0, MVF, and spectrum. Both F0 and MVF are scalars: F0 can be determined by any accurate method. The method used is a pitch detection algorithm that returns the MVF values at the analysis frames center. Cepstral coefficients are used to represent the spectrum. This distribution of Ahocoder contains two executable binary files built using GCC 4.4 under Linux (64bits): Coder translates waveforms into parameters; and Decoder translates parameters into synthetic waveforms [12]. There are voiced or unspoken types, to extract their cepstral information, and frames are treated differently. If the input frame was labeled as voiced, a harmonic is produced by the pitch detector. A harmonic analysis based on least squares is performed by the pitch detector. The complete analysis is subjected to squares optimization to obtain the harmonic amplitudes at various frequencies. These amplitudes are considered distinct, even at high resolution, samples of the real spectral envelope frequencies with a low harmonics-to-noise ratio. Unvoiced frames are subjected to a quick Fourier analysis (FFT), which is also known as a harmonic transform analysis with F0 equal to FFT resolution to be able to homogenize the discrete spectrum representation, the

harmonic amplitudes at voiced frames provide an envelope is resampled at the FFT after being normalized in amplitude interpolation for resolution. The Aho-coder includes linguistic processing and builds voices for some languages, such as English, Spanish, etc.

As a result, we can build a TTS framework with a feed-forward deep neural network (DNN) as shown in Figure 1.

B. Text-to-Speech with Limited Data and Multi-Language

FastSpeech2 [13] simplifies the training pipeline and overcomes the information loss as it is trained directly by a ground-truth target. Variation information of speech such as pitch, energy, and accurate duration are introduced to reduce the gap between input (text sequence) and target output (Mel-spectrogram) which reduces the one-to-many mapping problem. In the training phase the duration, pitch, and energy from the target speech wave-form are extracted as conditional inputs while in the inference, predicted values from the predictor are jointly trained with the FastSpeech 2 model. Using a continuous wavelet, the pitch contour is transformed into a pitch spectrogram which predicts the pitch in the frequency domain leads to improved accuracy of the predicted pitch [14]. FastSpeech2 is implemented with multi-language both English and Arabic.

As shown in Figure 2, phoneme embedding is converted using the encoder to phoneme hidden sequence, where the variance adaptor adds variance information such as pitch, energy, and duration into the phoneme hidden sequence, then the Mel-spectrogram decoder converts the adapted hidden sequence into Mel-spectrogram sequence in parallel. In training, the ground-truth value of duration, pitch, and energy is extracted into a hidden sequence to predict the target speech and is also used to train the duration, pitch, and energy predictors which infer to synthesized target speech.

After doing the baseline, our goal was to integrate the Arabic Language into FastSpeech2, the Arabic Speech Corpus was downloaded as a dataset, which is a modern standard Arabic for speech synthesis. It contains an orthographic and phonetic transcription of more than 3.7 hours of MSA speech aligned with recorder speech at the phoneme level, then the metadata was prepared and then trained the whole dataset

[15]. After it was implemented, we decreased the dataset to half 1.5 hours by adjusting the FastSpeech2 parameters, encoder, variance adaptor, and Mel-spectrogram decoder to match the different speaking speeds, loudness, tones, and timbre, to maintain a high-quality and a natural speech synthesis.

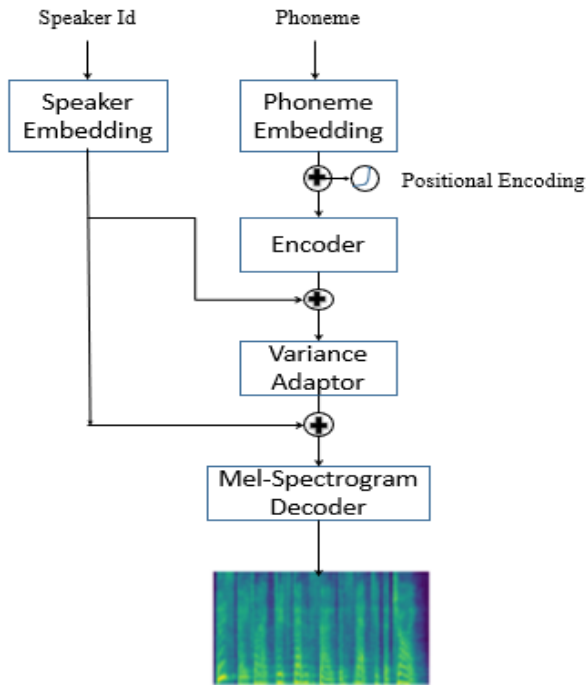


Figure 1: FastSpeech2 Architecture.

III. EXPERIMENTAL EVALUATIONS

Here we will discuss the different results that we got for each part of this experiment. It is divided into two parts: subjective results which are based on the preference and perceptual of the test participants which is the absolute category rating 5 being excellent and 1 being bad; and the Objective Results are based on the actual results that were gained from the training process.

A. TTS with Full Data

For the objective results, we evaluated three types of vocoders: WORLD, Continuous, and Ahocoder, as shown in Figure 3. The main goal was to integrate the Ahocoder and the continuous vocoder into the Merlin toolkit-based TTS. The advantage of the continuous vocoder is that it does not require the voiced or unvoiced decision, which reduces alignment error in the WORLD vocoder. Meanwhile, the Ahocoder has the advantage of providing accurate and high-quality speech synthesis and is well-suited for speech manipulation and transformation. The performance of the continuous vocoder was superior in most cases to that of the WORLD vocoder and Ahocoder. Additionally, it was found that the Ahocoder results system achieved slightly better scores than the WORLD vocoder.

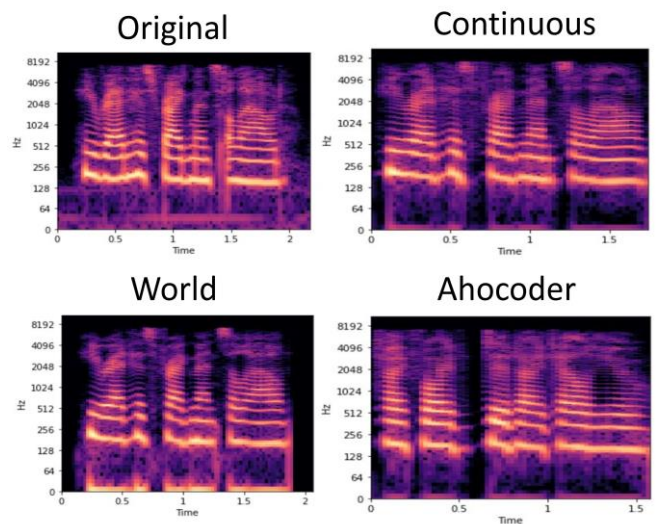


Figure 2: Results of three different vocoders.

For the subjective results, a listening test was conducted to compare the results for the different vocoders of our system. To evaluate the converted speeches, our test participants had to listen to the original voice of the full dataset, World vocoder voice, Continuous vocoder voice, and Ahocoder vocoder voice. The test participants had to rate the quality using the ACR scale., which refers to the absolute category rating, with 5 being excellent and 0 being poor. In Figure 4, we notice that the continuous vocoder was superior with high results close to the original source, followed by Ahocoder then the world vocoder.

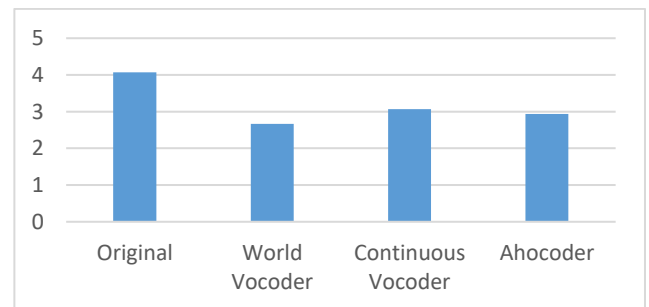


Figure 3: Sound quality of synthesized speech.

B. TTS with Limited Data

We conducted a listening test to compare the performance of the Arabic Text-to-Speech with Full Data and the Arabic Text-to-Speech with Limited Data. To evaluate the converted speech, our test participants listened to the original voice and the Arabic TTS and rated them using the ACR scale. The results, as shown in Figure 5, indicate that both the full data and limited data TTS were not comparable to the original sound.

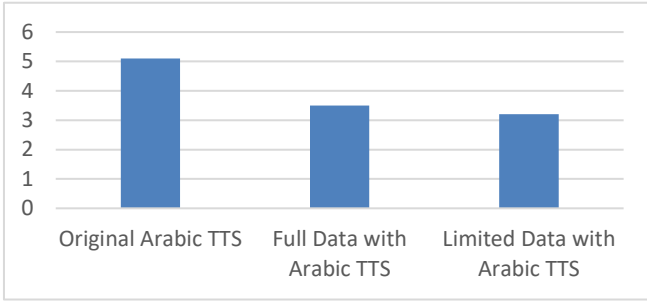


Figure 4: Subjective Results for the FastSpeech2 model using the Arabic Language.

For the objective results, we first implemented the baseline for the FastSpeech2 system, which only supported the English Language. We then integrated another language, Arabic Language, into the system. We also implemented it using less than half of the original dataset, while maintaining high-quality speech synthesis. This allows for the creation of a system where a user can train and generate speech using minimal data, which can be applied to more languages. Figure 6 shows the spectrogram results for the Arabic Text-to-Speech and the Arabic Text-to-Speech with limited data.

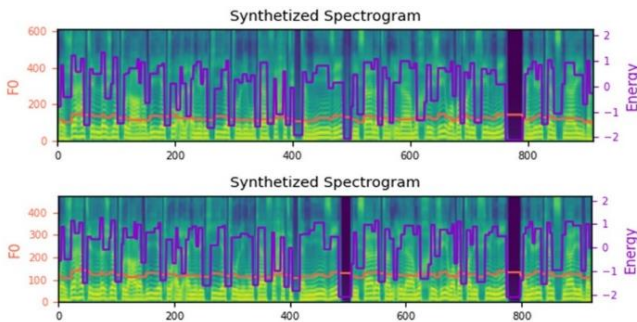


Figure 5: Objective Results of the FastSpeech2 model using the Arabic corpus: The top row shows results from the full data set, while the bottom row shows results from the limited data set.

In Table 1, we compare the results we got post-training for FastSpeech2 with limited and full data in the Arabic Language. We notice that the result for the limited data is still comparable to the full data synthesis.

Table 1: Training Results for the FastSpeech2 model using Full Data and Limited Data

Metrics	Full data	Limited data
Mel Loss	0.473	0.549
Mel PostNet Loss	0.472	0.549
Pitch Loss	0.331	0.906
Energy Loss	0.08	0.094

IV. CONCLUSIONS AND FUTURE WORK

This research proposed new approaches with the aim of developing a high-quality Text-to-Speech synthesis system focusing on the naturalness of the synthesized speech. We implemented different vocoders in TTS with full data to

investigate the effectiveness of various approaches in statistical parametric vocoders for speech synthesis quality. For TTS with limited data and multi-language, we used the state-of-the-art FastSpeech2 model, integrated the Arabic language, and then reduced the dataset to less than half of the original dataset. We obtained exceptional results. The performance of the systems was evaluated through subjective and objective tests. The results showed that the Continuous vocoder outperformed the other vocoders, followed by the Ahocoder, and then the World vocoder. The findings for TTS with limited data and multi-language indicate that our proposed model generates higher output speech quality than the baseline when integrating another language, as well as exceptional results when using limited datasets.

For future work, we plan to use a dataset with more diversified speakers and more languages. We will also focus on voice conversion and speech style to increase the output quality and enable the model to perform unseen speaker adaptation.

REFERENCES

- [1] Toda T.: Augmented speech production based on realtime statistical voice conversion. In: Proc. GlobalSIP, pp. 755–759 (2014).
- [2] B. Bollepalli, L. Juvela, and P. Alku, “Speaking style adaptation in text-to-speech synthesis using sequence-to-sequence models with attention,” arXiv.org, 29-Oct-2018. [Online]. Available: <https://arxiv.org/abs/1810.12051>.
- [3] Wu Z., Watts O., King S.: Merlin: An Open Source Neural Network Speech Synthesis System. In Proc. 9th ISCA Speech Synthesis Workshop (SSW9), Sunnyvale, CA, USA (2016).
- [4] Morise M., Yokomori F., Ozawa K.: WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE transactions on information and systems, vol. 7, no. E99-D, pp. 1877-1884 (2016).
- [5] Al-Radhi, M.S.: High-Quality Vocoding Design with Signal Processing for Speech Synthesis and Voice Conversion, Ph.D. Dissertation, BME University (2020).
- [6] Erro D., Sainz I., Navas E., Hernaez I.: Harmonics Plus Noise Model Based Vocoder for Statistical Parametric Speech Synthesis. IEEE Journal of Selected Topics in Signal Processing, vol. 8, no. 2, pp. 184-194 (2014).
- [7] Hu Q., Richmond K., Yamagishi J., Latorre J.: An experimental comparison of multiple vocoder types. In: Proc. ISCA SSW8, Barcelona, pp. 155-160 (2013).
- [8] M. S. Al-Radhi, T. G. Csapó, and G. Németh, “Noise and acoustic modeling with waveform generator in text-to-speech and neutral speech conversion,” Multimedia Tools and Applications, vol. 80, no. 2, pp. 1969–1994, 2020.
- [9] Garner P., Cernak M., Motlicek P.: A simple continuous pitch estimation algorithm. IEEE Signal Processing Letters, vol. 20, no. 1, pp. 102-105 (2013).
- [10] Drugman T., Stylianou Y.: Maximum Voiced Frequency Estimation Exploiting Amplitude and Phase Spectra. IEEE Signal Processing Letters, vol. 21, no. 10, p. pp. 1230–1234 (2014).
- [11] Tokuda K., Kobayashi T., Masuko T., Imai S.: Mel-generalized cepstral analysis - a unified approach to speech spectral estimation. In: Proc. ICSLP, pp. 1043–1046 (1994).
- [12] Erro, D., Navas, E., Sainz, I. and Hernaez, I.: Efficient spectral envelope estimation from harmonic speech signals. Electronics Letters, 48(16), pp.1019-1021 (2012).

- [13] Ren Y., Hu C., Tan X., Qin T., Zhao S., Zhao Z., Liu T.: FastSpeech2: Fast and High-Quality End-to-End Text to Speech. The International Conference on Learning Representations (ICLR) (2021).
- [14] Huang, S.-F., Lin, C.-J., Liu, D.-R., Chen, Y.-C. and Lee, H.: Meta-TTS: Meta-Learning for Few-Shot Speaker Adaptive Text-to-Speech. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 30, pp.1558–1571 (2022).
- [15] Halabi, N. and Wald, M.: Modern Standard Arabic Phonetics for Speech Synthesis (2016).