Deep Learning Models for Ukrainian Text to Speech Synthesis

Serhii Kondratiuk [a,b], Danylo Hartvih [c], Iurii Krak [a,b], Olexander Barmak [d], and Vladislav Kuznetsov [a]

[a] *Glushkov Cybernetics Institute, Kyiv, 40, Glushkov ave., 03187, Ukraine*
[b] *Taras Shevchenko National University of Kyiv, Kyiv, 64/13, Volodymyrska str., 01601, Ukraine*
[c] *Technical University of Munich, Arcis str. 21, München D-80333, Germany*
[d] *Khmelnytskyi National University, 11, Institutes str., Khmelnytskyi, 29016, Ukraine*

Abstract

The developed technology performs text to speech generation for the Ukrainian language. Implemented technology at first performs mel-spectrogram prediction using character sequence as an input and then based on composed acoustic features, the final audio signal is reconstructed. As a part of research, an overview and comparison of existing text to speech technologies for the Ukrainian language was conducted. For both stages deep learning architecture models were used – Tacotron2 for mel-spectrogram generation and Parallel WaveGan for voice audio generation. The research also contains a comparison with other architectures and vocoders, such as FastSpeech2 and WaveNet. Also, a grapheme-to-phoneme translation was improved in order to get better pronunciation quality for Ukrainian language. Experiments contain the usage of pre-trained models on English and other languages, in order to leverage transfer learning techniques. Research also contains a review and analysis of existing open-source datasets for the Ukrainian language. A mean opinion score metric was proposed and used in order to evaluate the final solution, with a statistically significant variety of generated testing samples and participants, based on defined criteria of the metric grade. Experiments show dependency on the training vocabulary of the dataset and further development implies augmentation of the dataset with different topics. The best configuration for all deep learning models was found based on training and testing results are also shown in the research, with hardware requirements needed to reproduce the training. Experiments also show the satisfying quality of generated voice, trained on a specially collected and processed dataset of the single male voice. Testing experiments 20 people participated, while each person evaluated about 100 generated sound samples of length from 40 to 60 seconds score of 4.02 of mean opinion score metric was achieved.

Keywords [1]

Text to speech, deep learning synthesis, tacotron2, mel-spectrogram, parallel wavegan

1. Introduction

Text to speech has lots of applications, it is a widespread technology, which can be used to help people with a wide range of disabilities. They can also be used in entertainment production, to make voice acting production cheaper. Multiple high-quality text-to-speech solutions exist, such as [1], [2], however, they commonly provide high quality for very widespread languages. Speech generation for less spread languages, such as Ukrainian, is a harder task, whilst the demand is only growing. It's important to provide an open-source framework that contains a pipeline to train new voice models for the Ukrainian language in a convenient way. Also, it's vital to prove that the pipeline is providing high-quality results with test voice samples. Let note that for more effective speech generation need

[1]ntellTSIS'2023: 4th International Workshop on Intelligent Information Technologies & Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: sergey.kondrat1990@gmail.com (S. Kondratiuk); danil.gartvig@gmail.com (D. Hartvih); iurii.krak@knu.ua (I. Krak); alexander.barmak@gmail.com (O. Barmak); kuznetsow.wlad@gmail.com (V. Kuznetsov)
ORCID: 0000-0002-5048-2576(S. Kondratiuk); 0000-0002-4593-411X (D. Hartvih); 0000-0002-8043-0785 (I. Krak); 0000-0003-0739-9678 (O. Barmak); 0000-0002-1068-769X (V. Kuznetsov)

investigate cognitive linguistic analysis [3], including of phonetic analysis and grammar structure of language [4]-[6], speech signals marking and segmentation [7].

There are only a few existing solutions for Ukrainian language Text to Speech (TTS) generation. The majority of them [8]-[10], are formant synthesis models (so-called rule-based synthesizers). Those approaches have the advantage of low resource usage and high-speed synthesis, at the cost of generating artificial, robot-like speech. They are commonly used as screen readers for visually impaired people.

There are also two deep learning solutions for the Ukrainian language: WaveNet-based service from Google Could Text-to-Speech [11] and Nuance Vocalizer TTS [12]. However, they are proprietary and it's not possible to train a new voice with those cloud providers.

Speech synthesis from the text in a single end-to-end step is a very sophisticated task, such a model would be hard to train and interpret. Commonly synthesis is split into two stages. The first stage is to train a synthesizer (which is also known as Encoder-Decoder architecture) on character sequence in order to predict mel-spectrograms [13] (a low-level acoustic representation). The second stage is to train a neural vocoder (waveform generator) model, which uses acoustic features from the previous step to reconstruct audio signals (final voice sample). In order to get a reduced dimensionality view, we opted to make use of short time Fourier transform, or also known as STFT. This applies to some data, for instance, audio or other signals. Using such non-linear transform, we can view in detains the underlying nature of the data, focusing mostly on low frequency and more pronounced details of the signal, rather than high-frequency details, that may contain some artifacts due to discretization process or size of the window.

Thus, the purpose of this work is to develop information technology for the generation of text to speech for Ukrainian language. A technology is proposed that, using a sequence of speech symbols, performs the prediction of mel-spectrograms and, based on the compose acoustic characteristics, transform it into a voice signal. To implement these stages of the research, it is proposed to use modern models of deep learning architecture - Tacotron2 for the generation of a mel-spectrogram and Parallel WaveGan for the generation of voice audio. Comparison with other architectures and vocoders such as FastSpeech2 and WaveNet showed better results of the proposed approach. Experiments show the dependence on the training dictionary of the data set, and further development involves supplementing the data set with various topics. Grapheme-phoneme translation has also been improved for better pronunciation quality in Ukrainian. The structure of the article consists of the following sections: introduction; an overview of existing approaches to machine learning of textual information using different architectures of deep learning models, indicating their advantages and disadvantages; detailed description of the proposed approach; description of datasets for training procedures; conducting experiments and analyzing the obtained results; conclusions with an assessment of the obtained results and plans for further research.

2. Related Works

Predicting audio signals directly from text with one step is a complex and sophisticated task. That's why deep learning synthesis is splinted into two stages. In the first stage, the synthesizer used preprocessed character sequences to predict mel-spectrograms (a low-level acoustic representation). In the second stage, the neural vocoder model uses acoustic features (mel-spectrograms) to reconstruct audio signals.

We used a STFT on a scale of 50ms along the studied data. The parameters passed to the method are the following: hop is 12.5 milliseconds, the used function to display the spectrogram is Hann (you can see the overall results on the Fig. 1). We also used an additional transformation, so as the spectrogram was scaled in the logarithmic scale with 80 channels in order to achieve a proper mel-cepstral diagram of the input data.

There is commonly known fact, that humans perceive pitch (either musical or sound) differently in low and high range of the perceived scale. The lower sound (or "note") is easily distinguishable from the higher, so as the lower the pitch, the higher the ability to distinguish two pitches. For instance, humans would likely distinguish tones of heavy electric machinery (which lay in relationship times per feeding electric grid frequency which is 50/60Hz), from the sounds of electric pulse modulation electric

AC-DC adaptors (some of them may have a range well beyond the 10000+ Hz, that can't be heard by elderly people or people with some hearing problems).

Based on thoughtful study, there was a solution found in order to organize a sound spectra. Circa 1937, a team of scientists including Stevens, Volkmann, as well as Newmann suggested a metric in order to measure distances between different pitches of the sound spectra. In order to do so they studied, how the different pitches relate one to another to a human. This metric, now commonly known as mel-scale (see Fig. 2), allows to plot the frequencies in such manner as the human would plot them; it also defined an easy forward and backward transformation so as it would be possible to convert STFT diagram to a MEL diagram and vice versa.

In the proposed research we perform a mathematical operation on frequencies to convert them to the mel scale.

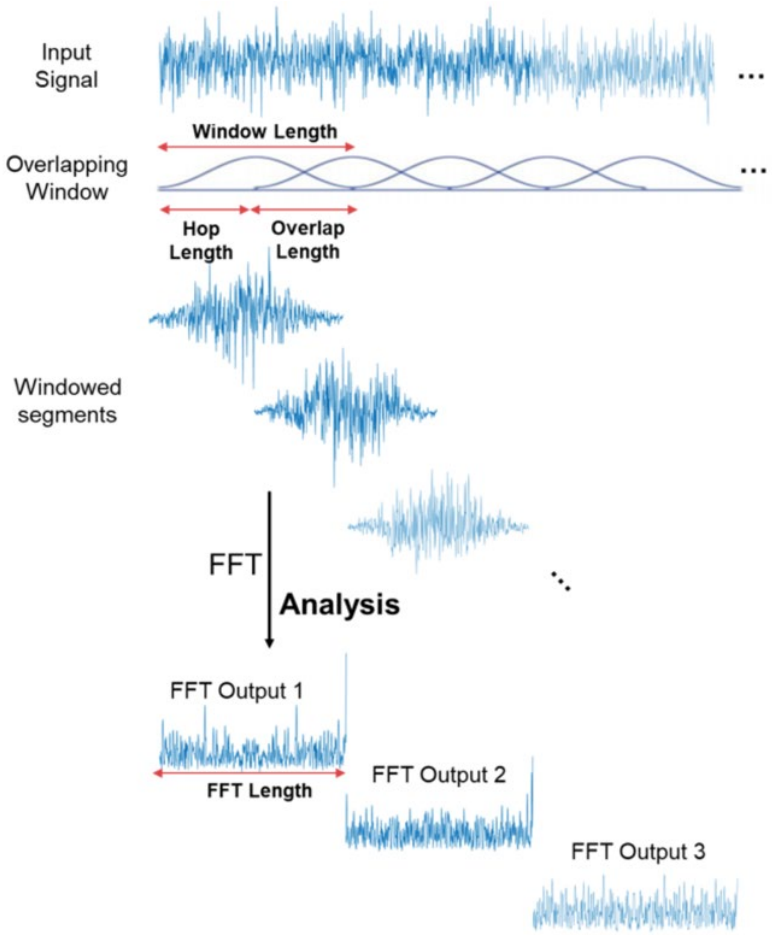

Figure 1: Fast Fourier transformation visualization

Logarithmic transformation of the spectrogram into a mel-scale is well known approach or technique for sound signals dimensionality reduction. Despite the fact it neglects some information of the signal in the complex time-frequency scale, it is useful for some denoising techniques, and hence can find various areas of usage.

We must emphasize, since the inverse transformation exists, the result of the backward transformation is not equal to the original signal, since the transformation is lossy. To overcome this problem, different approaches were proposed; for instance, in papers [14,15] one can find an approach of using Griffin Lim or neuro voice-coders in order to obtain more accurate representation from a raw signal.

The mel spectrogram, with applied range of techniques discussed above, can be seen on Fig. 3.

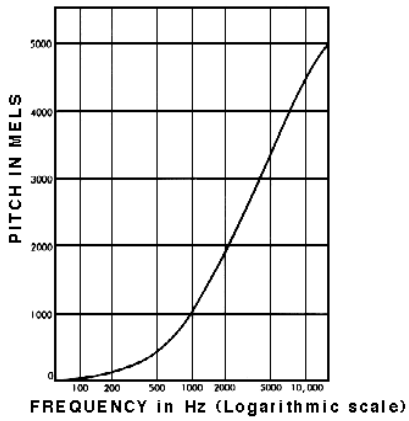

Figure 2: Mel scale

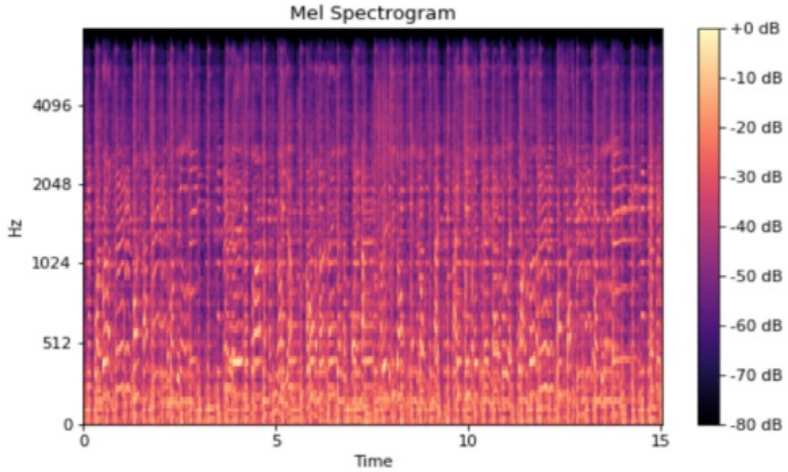

Figure 3: Mel spectrogram

Two (most popular/state-of-the-art) synthesizers are Tacotron2 [16] by Google and Fastspeech2 [17] by Facebook.

Tacotron2 is an encoder-attention-decoder neural network, where encoder converts input sequence to a hidden feature representation. (Encoder has 3 convolutional layers and Bidirectional Long Short-Term Memory (LSTM)). After that attention summarizes the encoded sequence to a fixed-length vector which the decoder consumes to predict audio features (mel-spectrogram). (Decoder is an autoregressive recurrent neural network used with 2 Fully Connected layers as Pre-Net and 5 Convolutional Layers as Post-Net).

Since the model is only capable of predicting fixed-length output, in parallel it also predicts stop-token which allows the model to terminate generation earlier.

Original Tacotron2 is a single-speaker model, meaning that one model is able to generate only one voice. But there is an improved version of Tacontron that has the speaker's voice embedding as second input [21]. Tacotron architecture is shown at Fig. 4.

Let's discuss the FastSpeech architecture more in detail. It consists of transformers, as well as encoder-decoder blocks. This architecture has a variance adaptor block, as well as 4 transformer sub-blocks within each encoder and decoder.

The main purpose of an encoder is to transform the data from the original dimension into a hidden one.

The variance module can be used, in order to process such complex signal as speech, using different parameters, for instance pitch, duration, energy density and others.

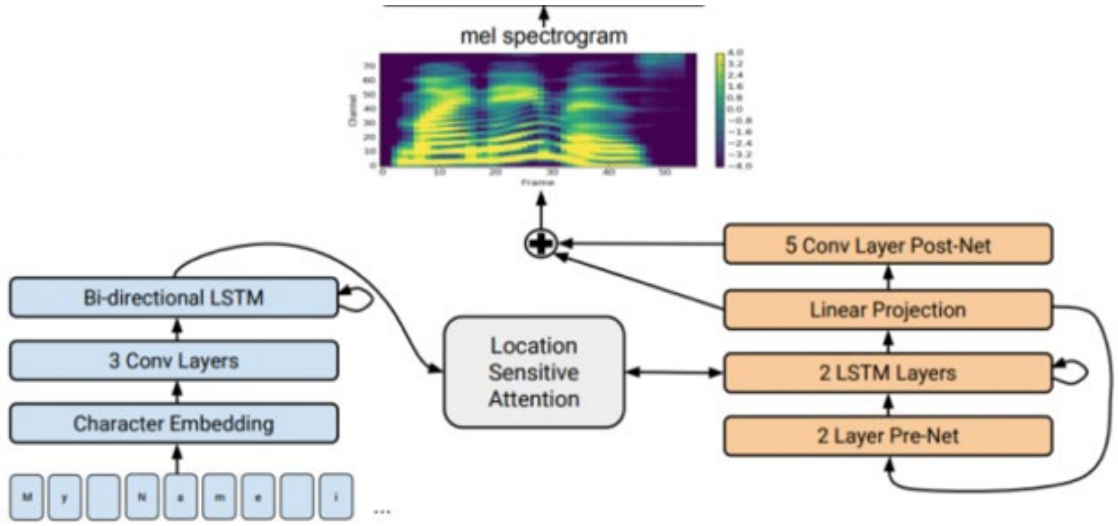

Figure 4: Tacotron architecture

In order to correctly reconstruct the original signal from the hidden transformation, the decoder learns the hidden features and the inverse transformation function as well, using a variance module as well.

There are many implementations and modifications of this method, including FastPitch, as well as MultiSpeech; while being based on FastSpeech architecture, they differ in purposes - the first is more suited to pitch prediction, while the latest for utterances that have multiple participants.

You can see the architecture of FastSpeech more in detail on Fig. 5.

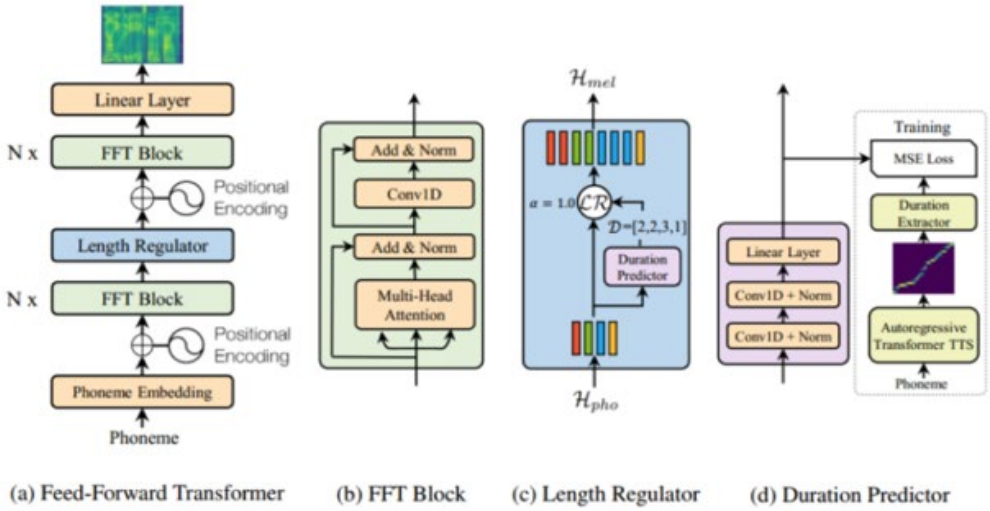

Figure 5: Fastspeech2 architecture

Let's emphasize on FastSpeech advantages, based upon the provided Fig.5. Firstly, it has a straightforward architecture, consisting of an encoder and decoder, as well as transformer, that learn the transformation of data, minimizing the error. Despite the fact that such architecture found in autoencoders may eliminate some information due to denoising effect of autoencoders in general, it may be helpful and precise enough for tasks of sound and speech analysis.

We also want to emphasize, that this architecture is used not only to learn the transformation not only to transform the signal into a hidden feature-space, but also to perform fast Fourier transforms along the input signal. Hence, in order to perform such a task, the transformer makes use of self-attention layer, as well as convolution. The arrangement of the blocks is dictated by the level of detail of features on each layer of the network. While the layer closer to the input encode the low-level pitch

features, the latest encode the features of phonetic features and specgram level as well. In order to adapt to a speech, one can fine tune hyperparameters so as to adapt better to different length of the utterances and phonemes as well.

In order to further discuss the architecture, let's focus on some of its parameters, depicted above. Since a phoneme can be encoded by multiple pieces of the specgram, in general the utterance is generally shorter than a sequence itself. Hence, the length hyperparameter restricts the overall number of specgrams used to encode each phoneme. This parameter is also used to implement dynamic transformation of timescale of the hidden sequences so as they correspond to the desired length. One way to control this parameter is to check the speed of the speech per fixed amount of time so as it can control the length parameter "on the fly". It can also be used to segment unnecessary parts of the signal, that likely don't contain useful signal. The key idea between this technique is an automatic regulator, that feeds the estimated length of each phoneme and hence controls the "speed" of the transformation, which obviously won't work in other cases, if it was a fixed value. The regulator relies upon a block, consisting of conv and shallow layer, as it shown above. The predictor for a sequence regulation is trained upon the error ratio and is connected to the FFT transformation block (on the bottom side, left image). Hence, we can say, that the model, being controlled by a speaker tempo, relies strongly upon the attention between the forward and inverse encoding parts of the model.

3. Proposed technique

We need to underscore some problem may happen during the text conversion phase, if the speaker is badly heard, especially, in noisy environment. Sometimes the phonemes in the sequence can be mistaken with others, for instance the transition from one sound into another. We suggest treating this situation carefully, since sometimes the same letters (written) can have similar sound or corresponding phoneme. Hence, it is very important to catch not only the raw transcription, but being able to interpret if there can be a margin for an error, like mispronunciation.

The phoneme does not have an independent lexical or grammatical meaning but serves to distinguish and identify significant units of the language (morphemes and words). In the Ukrainian language, there are 33 graphemes, of which 21 are consonants and 12 vowels. The phonetic system of the modern Ukrainian language has 38 basic phonemes: 6 vowels and 32 consonants; additionally, determines 10 double consonant phonemes in the peripheral subsystem.

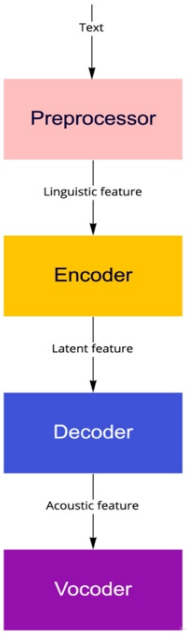

Figure 6: TTS system architecture

We used Encoder-Decoder Recurrent Neural Network (RNN) [19] with Attention [20] used to convert grapheme representation of words to phonemes. Encoder-Decoder RNN is usually used for machine translation, and since the grapheme to phoneme conversion task is similar to translation, it is

naturally a good solution. Encoder and Decoder consist of one layer each with bidirectional LSTM and hidden dimension size of 256, while decoder part also has an Attention layer. High-level architecture of our system is showed at Fig. 6. Let's discuss the limitations of the model. Obviously, one should be aware of the fixed length of the input audio sequence. Since the model is affected by attention between the forward and inverse encoding, as we noted before, it is also important to carefully process the utterances of different length. Since the regulator adapts not only to the timescale of the speaker, it also affects the timescale of the utterance in some manner, which may be a difficult problem to overcome. Hence, we could discover a problem where we speak about same utterances, but having absolutely different timescale and hence some phonemes may be represented by various number of the sequences, even if they represent the same phrase. As a consequence, we suggest addressing this problem with detail in both the task of decoding of sequences, with same level of attention, as it done in more complex tasks, such as text-to-text or audio-per-audio translation.

Fig. 7 shows attention layers used in the technology.

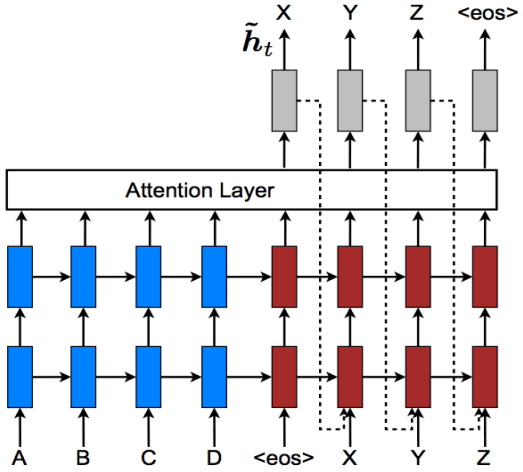

Figure 7: Attention layer

What are the main differences of Tacotron2 from the discussed here Fastspeech2 architecture in terms how it fits everything up? Obviously, the key is what it drives up: while the Tacotron benefits from autoregression to generate specgrams, the Fastspeech relies on the self-attention in FFT blocks used in the forward and inverse encoding [18]. As consequence, we must remember that the FastSpeech2 is not all-way fit method: it should be trained with more attention, hence it has carefully to predict the duration of the segments, but also to define the specgrams, calculated from the input audio; if the duration is not predicted correctly, we can observe the effect of the overlapping, which may drastically decrease the accuracy of the model. As shown in [21], Fastspeech2 overcomes some controllability issues of Tacotron2, however, their metrics show comparable performance. In implemented technology, we ended up with Tacotron2 because it was easier to integrate with other parts of the pipeline. Since the model is only capable of predicting fixed-length output, in parallel it also predicts stop-token which allows the model to terminate generation earlier. Original Tacotron2 is a single-speaker model, meaning that one model is able to generate only one voice. There is an enhanced version of Tacontron that has the speaker's voice embedding as second input, which leads to a minor drop in quality, in this research we ended up using the original Tacotron2 implementation.

One of the problems with Tacotron2 is the instability of the gate layer, which is responsible for stopping generation, and if it's not working properly, the model decoder will continue to generate mel-spectrogram frames until it reaches the limit (max-length). To solve this problem, we add the End Of Sentence (EOS) symbol at the end of each input sequence. Another problem is the instability of the mechanism of attention in the original implementation of the Tacotron2.

In order to overcome the issue we suggest the following approach: at first, we apply the dial guided attention, so as the model can learn faster. In order to control the training process, we experimentally figured out the desirable number of iterations (in the literature, one can find that this number is varying and likely around, 20000). As a result, the penalty on the representation attention matrix is made when it is distinct from a diagonal. We utilized, as a part of text-to-speech synthesis system, a voice coder-decoder, which generates the audio from the specgrams. Using such models, as proposed by Griffin

Lim, we can generate a transformation from an inverse (MEL) representation to a raw audio data (which was not possible in early implementations of logarithmic transformations, due to loss of complex phase from the STFT specgram) using a network to study it from data. Also, we want to emphasize, that instead of using decoder by itself, we can proceed to it via latent feature space representation and then to audio. We tried various types of voice coders-decoders, including MelGan, Hi-Fi GAN, WaveNet [22-25]; however, none of them did not fully answer our needs. As a result, we ended up with Parallel WaveGan. The architecture of it is depicted on the Fig. 8.

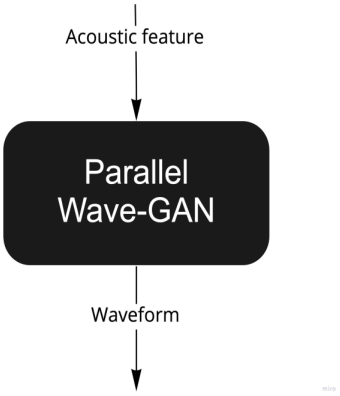

Figure 8: Vocoder schema used

In order to perform our task, as we noted before, we used Parallel WaveGAN. The main benefit of this method is utilization of relatively small GAN, that is always good if the system makes use of a lot of time- and memory- consuming libraries. Hence, using such method, we can avoid the obvious drawbacks we noticed in Tacotron2 architecture and benefit from loss functions, that can better represent the features of the human speech, such as distribution of the voice information within the output signal, as well as using different resolutions of specgrams. We can see, as other positive effect is density distillation, which is not used in other models we discussed before.

4. Datasets

Tacotron2 training required dataset in special format LjSpeech [26]: length of audio segment from 2 to 20 seconds and corresponding transcription for all audio files. There is only one opensource dataset for Ukrainian language that has the correct format for Text-to-Speech task: M-AILABS Speech Dataset. M-AILABS Speech Dataset consists of audio books splitted in short segments with transcription, it has several speakers but for Tacotron2 training we choose only one with the best audio quality and at least 15 hours of data. Moreover, we preprocessed all audio: noise reduction performed with Fourier Transformation, volume level normalization, conversion to mono channel and 22050 Hz sampling rate. Split the dataset into 3 parts: N samples for training with total duration N, M samples (M hours) for validation and L samples (L hours) for testing.

We transform transcriptions into the phoneme sequence using the model described above (Grapheme-to-Phoneme [27] translation). Since we predicting mel-spectrograms, raw audios also were transformed.

To train universal Vocoder we used the opensource dataset Common Voice [28] which also has Ukrainian language. It is a multi-speaker dataset created by Mozzila, it has 615 unique voices and 56 hours of validated data. Also, for better results we mixed Common Voice with 10% of audios from the dataset used for Tacotron2. All audios have been converted to 22050 Hz sampling rate. Split the dataset into 3 parts: N samples for training with total duration N, M samples (M hours) for validation and L samples (L hours) for testing. We transformed raw waveforms (audios) into mel-spectrograms, which is input data to the vocoder model.

5. Experiments, results and discussions

Training and inference were performed using a Google Colab [29], using such hardware:

- GPU: Nvidia P100 16GB
- CPU: 2x Intel Xeon CPU @ 2.20GHz
- RAM: 13 GB

Tacotron2 was trained for 10 days, with 125,000 iterations and batchsize 32 with next architecture:

- emb_hidden: 512
- encoder
    a. 5 conv layers
    b. lstm: 256 units
- decoder
    a. 2 prenet layers
    b. 1 lstm
    c. lstm units: 1024
    d. attention dim 128
    e. postnet with 5 conv layers

Fastspeech2 was trained for 35 hours, with 200,000 iterations and batchsize 32 with architecture:

- emb_hidden: 384
- encoder
    a. 4 hidden layers
    b. 2 attention heads
- decoder
    a. 4 hidden layers
    b. 2 attention heads
- variant prediction
    a. 2 conv layers
    b. filter 256
    c. kernel size 3
    d. dropout 0.5

Parallel WaveGAN was trained for 5 days approximately, with 600,000 iteration and batchsize 16. Training charts (total loss and regularization loss) are shown at Fig. 9.

During experiments, in contrast to training, validation and test metrics during training, it is hard to evaluate TTS engine with some strict metric, due to only human can truly evaluate if the output sound meets couple of requirements:

- natural sound / no “robotic” sounding
- no noise
- no interruptions

In order to generate scores for our solution, we utilized commonly used metric - Mean Opinion Score (MOS). This metric is commonly used in various applications, which in simple words can be compared to the grades in school - from 1 to 5. Hence, we use this simple metric to evaluate the average for different parameters of the model. We must also underscore that this metric nowadays strongly relies upon the opinions of the experts in the area, it is a quite reasonable method to get approximate ranking; hence we utilize commonly used ratings varying from 5 (Excellent) to 4 (Good), 3 (Fair), 2 (Poor) and 1 (Bad) as in average category ranking scale (ACR).

Since we can't always achieve the highest rank (5) due to variability in ranks, we consider that every rank over 4.3 can be considered enough to be named excellent. We also want to exclude poor quality examples, so as we don't consider any below 3.5.

During our experiment, using a group of 20 participants, we asked each one to rank 100 sequences, which are described more in detail in Table 1; in overall we achieved score 4.02, which is quite good for our tasks, according to metric.

Based on feedback from test group, and overall testing results, such insights were obtained:

- Dependency on the training vocabulary, since the technology was trained on fiction texts, news texts were synthesized with a generally worse quality;
- Pretrained weight produced good impact on training process. Pretrained models on other languages provided better faster convergence for Ukrainian language.

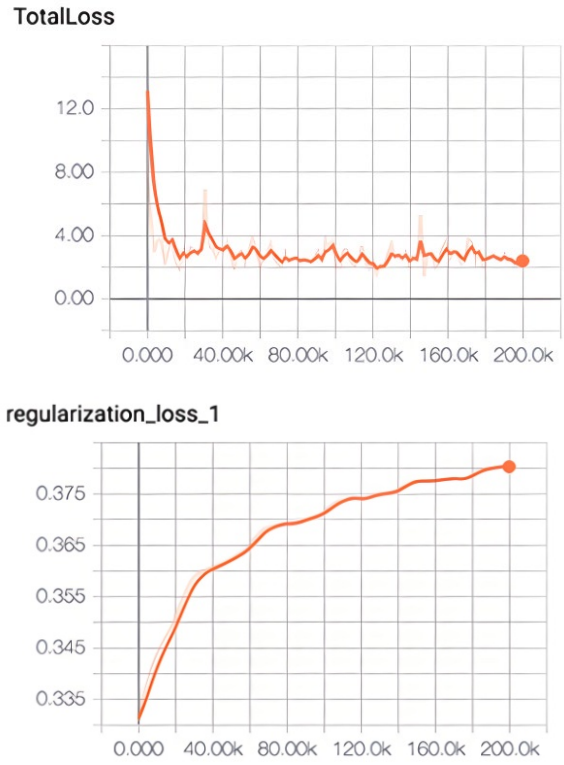

Figure 9: Training charts

Table 1
Details about testing statistics

Characteristic	Value
People participated	20
Evaluated samples	2000
In minutes	1500
Highest MOS per person	4.6
Lowest MOS per person	3.8

6. Conclusions

As a result of the work, a novel engine for text to speech synthesis for Ukrainian language was proposed and implemented. The accuracy of the method was also assessed. Architecture consists of two main parts (encoder/decoder and vocoder), and smaller preprocessing part with phonemes. A novel approach to Ukrainian language phones was presented and implemented as a part of technology.

During implementation multiple approaches and models were considered, such as Tacotron2, Fastspeech2, Parallel Wave GAN, MelGan, Hi-Fi GAN, WaveNet and other. Based on experimental results, Tacotron2 and Parallel Wave GAN were selected. A proper metrics (MOS) was proposed and measured as a part of work. A statistically significant testing was performed using 20 people and a MOS of 4.02 was achieved. Among the limitations of the proposed approach, it should be noted that significant computing resources should be used for implementation.

Further development of the text to speech engine will address improvement its versatility based of the dataset.

7. References

[1] Amazon AWS: Polly. URL: https://aws.amazon.com/en/polly/

[2] Microsoft Azure: Text to speech. URL: https://azure.microsoft.com/en-us/services/cognitive-services/text-to-speech/
[3] Kovtun V., Kovtun O. System of methods of automated cognitive linguistic analysis of speech signals with noise, Multimedia Tools and Applications. Springer Science and Business Media LLC, 2022. https://doi.org/10.1007/s11042-022-13249-5
[4] Kovtun V., Kovtun O., Semenov A. Entropy-Argumentative Concept of Computational Phonetic Analysis of Speech Taking into Account Dialect and Individuality of Phonation, Entropy, vol. 24, no. 7, 2022; 1006. https://doi.org/10.3390/e24071006
[5] Krak, Y., Barmak, O., Mazurets, O. The practice implementation of the information technology for automated definition of semantic terms sets in the content of educational materials. In: CEUR Workshop Proceedings 2139, pp. 245-254 (2018). doi:10.15407/pp2018.02.245
[6] I.G. Kryvonos, Iu.V.Krak, O.V.Barmak, R.O. Bagrii. New Tools of Alternative Communication for Persons with Verbal Communication Disorders. Cybern. Syst. Anal. 52(5), 655–673 (2016). doi: 10.1007/s10559-016-9869-3
[7] Rashkevych, Y., Peleshko, D., Pelekh, I., Izonin, I. Speech signal marking on the base of local magnitude and invariant segmentation. Mathematical Modeling and Computing, 2014, 1(2), pp. 234–244.URL: https://ena.lpnu.ua/handle/ntb/26455
[8] UkrVox. URL: https://biblprog.org.ua/ru/ukrvox/
[9] URL: https://gud.rv.ua/
[10] RHVoice. URL: https://rhvoice.su/
[11] Google Cloud: Text to speech. URL: https://cloud.google.com/text-to-speech
[12] Cerence/Nuance TTS Ukrainian. URL: https://nextup.com/cerence/
[13] Mel-spectrogram. URL: https://en.wikipedia.org/wiki/Mel_scale
[14] Griffin-Lim Algorithm. URL: https://paperswithcode.com/method/griffin-lim-algorithm
[15] J. Yu, et al. DIA-TTS: Deep-Inherited Attention-Based Text-to-Speech Synthesizer. Entropy. 2023; 25(1):41. https://doi.org/10.3390/e25010041
[16] J. Shen, et al. TTS Synthesis by Conditioning Wavelet on Mel Spectrogram Predictions. URL: https://arxiv.org/pdf/1712.05884.pdf
[17] Y. Ren, C. Hu, X. Tan, T. Qin. FastSpeech2: Fast and High-quality End-to-end Text to Speech. URL: https://arxiv.org/pdf/2006.04558.pdf
[18] Y. Ren, et al., Fastspeech: Fast robust and controllable text to speech, Advances in Neural Information Processing Systems. URL: https:// proceedings.neurips.cc/paper/2019/ file/f63f65b503e22cb970527f23c9ad7db1-Paper.pdf
[19] K. Cho, at al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. URL: https://arxiv.org/abs/1406.1078
[20] D. Bahdanau, K. Cho, Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. URL: https://arxiv.org/abs/1409.0473
[21] Tacotron2 vs FastSpeech2 – URL: https://towardsdatascience.com/text-to-speech-with-tacotron-2-and-fastspeech-using-espnet-3a711131e0fa
[22] K. Kumar, at al. MelGan: Generative Adversarial Networks for Conditional Waveform Synthesis URL: https://arxiv.org/abs/1910.06711
[23] J. Shen, et al. Natural TTS Synthesis by Conditioning Wavenet on MEL Spectrogram Predictions, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*)*, Calgary, AB, Canada, 2018, pp. 4779-4783, doi: 10.1109/ICASSP.2018.8461368
[24] J. Kong, J. Kim, J. Bae. HIFI-GAN: Generative Adversarial Networks for Efficient and High-Fidelity Speech Synthesis. URL: https://arxiv.org/abs/2010.05646
[25] A. Oord, at al. WaveNet: A Generative Model for Raw Audio URL: https://arxiv.org/abs/1609.03499
[26] K. Park, T. Mulc. CSS10: A collection of single speaker speech dataset for 10 languages. URL: https://arxiv.org/pdf/1903.11269.pdf
[27] S.O. Arik, at al. Deep Voice: Real-time Neural Text-to-Speech. Proceedings of the 34th Intern. Conf. on Machine Learning, 70:195-204. 2017. https://proceedings.mlr.press/v70/arik17a.html
[28] Mozilla Foundation: Common Voice Dataset. URL: https://commonvoice.mozilla.org/en/datasets
[29] Google Research: Google Colab. URL: https://colab.research.google.com/