applied sciences

MDPI

Article

MAKEDONKA: Applied Deep Learning Model for Text-to-Speech Synthesis in Macedonian Language

Kostadin Mishev [1,†], Aleksandra Karovska Ristovska [2,†], Dimitar Trajanov [1,†], Tome Eftimov [3,†] and Monika Simjanoska [1,*,†]

[1] Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, 1000 Skopje, North Macedonia; kostadin.mishev@finki.ukim.mk (K.M.); dimitar.trajanov@finki.ukim.mk (D.T.)

[2] Faculty of Philosophy, Ss. Cyril and Methodius University, 1000 Skopje, North Macedonia; aleksandrak@fzf.ukim.edu.mk

[3] Computer Systems Department, Jožef Stefan Institute, 1000 Ljubljana, Slovenia; tome.eftimov@ijs.si

[*] Correspondence: monika.simjanoska@finki.ukim.mk

[†] These authors contributed equally to this work.

Received: 9 September 2020; Accepted: 24 September 2020; Published: 1 October 2020

Abstract: This paper presents MAKEDONKA, the first open-source Macedonian language synthesizer that is based on the Deep Learning approach. The paper provides an overview of the numerous attempts to achieve a human-like reproducible speech, which has unfortunately shown to be unsuccessful due to the work invisibility and lack of integration examples with real software tools. The recent advances in Machine Learning, the Deep Learning-based methodologies, provide novel methods for feature engineering that allow for smooth transitions in the synthesized speech, making it sound natural and human-like. This paper presents a methodology for end-to-end speech synthesis that is based on a fully-convolutional sequence-to-sequence acoustic model with a position-augmented attention mechanism—Deep Voice 3. Our model directly synthesizes Macedonian speech from characters. We created a dataset that contains approximately 20 h of speech from a native Macedonian female speaker, and we use it to train the text-to-speech (TTS) model. The achieved MOS score of 3.93 makes our model appropriate for application in any kind of software that needs text-to-speech service in the Macedonian language. Our TTS platform is publicly available for use and ready for integration.

Keywords: macedonian language; text-to-speech; deep learning; natural language processing; speech synthesizer

1. Introduction

Text-to-speech (TTS) is a challenging problem has attracted researchers' attention over the past 30 years. When considering the literature, it can be easily perceived that the level of success is, however, determined by the evolution of methods that overcome the human abilities of analysing and extracting features. Instead, those methods provide a higher-level abstraction of characteristics that fill the gap of producing human-like speech—something that was missing in the early efforts. Generally, a traditional TTS system is comprised of two consecutive main parts: text analysis and speech synthesis. The text analysis part includes text processing, or, morphological analysis and rule-based syntax analysis. The speech synthesis part encompasses methods that are able to recreate a speech by using a defined set of rules. The main problem with the traditional models for achieving the text-to-speech translation is that they rely on one-layer nonlinear transformation units [1]. Those are Hidden Markov models (HMMs) [2–5], maximum Entropy based methods [6–8], and concatenation based synthesis methods [9–11].

It can be concluded that the existing attempts to create Macedonian TTS system, mostly rely on concatenation based synthesis methods, i.e., the synthesis is based on selecting an appropriate speech unit from a prerecorded and labeled speech corpus and adjusting the prosody of the concatenation unit according to the target context, as presented in the following Section 2. This approach quality is limited by the pitch period, or, starting point and the maintenance of smooth transitions [1].

In this paper, the focus is on creating the first Macedonian TTS system by following a parametric speech synthesis that is based on Deep Learning approach. The parametric speech synthesis has shown to significantly improve the naturalness of the generated speech [12–14]. Those approaches include three phases: text analysis, parameters prediction, and speech synthesis. All three steps have their traditional and DNN-based shape. The advantage of the DNN-based methods over the traditional ones is that they not only transform complex linguistic features into acoustic feature parameters, but also model the correlation between frames by using long short-term context information that improves the quality of the synthesized speech [1]. When compared to the HNN-based approach, in which the linguistic features are first mapped into probability densities, the DNN-based method directly maps the linguistic features to acoustic features.

The text analysis itself can be done at few different levels: phoneme level, syllable level, word level (analysis of part of speech—POS), phrase level, and sentence level.

Among the earliest attempts to use the leverage of DNN in the field of TTS is presented in [15]. The authors try to eliminate the decision tree clustering weakness (division of model space considering one feature at the time) in statistical parametric speech synthesis, by combining vector-space representations of linguistic context and DNNs.

One of the most famous and widely used is Tacotron, which is a seq2seq model [16], and its successor Tacotron2 [17,18]. Given (text, audio) pairs, Tacotron is able to directly synthesize speech from characters. Tacotron has been an inspiration for some newer approaches that successfully overcome the speed problem, producing close, or, equal quality speech. Such is FastSpeech [19] and FastSpeech2 [20], the newest TTS model published at the time. However, both autoregressive models depend on additional models for duration prediction, which means that Tacotron2 model has to be trained to some satisfactory accuracy and, therefore, used for duration prediction needed for training FastSpeech2. Another option is using Montreal Forced Aligner (MFA) in order to obtain the alignments between the utterances and the phoneme sequences.

Some valuable work done in the DNN-based speech synthesis field is presented in [21–23]. The papers present a consecutive progress of DNN-based synthesizers, more precisely, Deep Voice 1 and 2 [21,22] retain the traditional structure of TTS. They aim at separating grapheme-to-phoneme conversion, duration and frequency prediction, and waveform synthesis. Deep Voice 3 [23] is more complex character-to-spectrogram architecture, and it employs an attention-based sequence-to-sequence model. However, this approach proved to be most valuable for the successful creation of the Macedonian TTS system. Details can be found in Section 3.3.

Researchers are not only satisfied with the accuracy of the models, but also of the time-performance and resources-creed of the methods. The newest proposals deal with the quality-training time trade-off. Therefore, in [24], the authors describe a TTS technique that is based on deep convolutional neural networks (CNN), but without any recurrent units, instead, stressing the importance of training the attention module. This means that they propose a solution that will reduce the need of very powerful computing resources and also reduce the time that is required to run the experiments, which usually takes several weeks. In the experiment, they have done comparative analysis of whether the CNN-based algortihm will produce acceptable quality of speech. They have trained the model in 15 h by using an ordinary gaming PC that is equipped with two GPUs, and the results proved the efficiency of the network in terms of synthesized speech quality.

Some of the newest work is presented in [25]. In this work, the authors propose a new solution for parallel wave generation by WaveNet. The improved method in comparison to the previous work (Parallel WaveNet [26]), in which end-to-end speech synthesis actually refers to the text-to-spectrogram models with a separate waveform synthesizer (vocoder); this solution simplifies the training algorithm by introducing the first text-to-wave fully convolutional neural architecture for speech synthesis, which enables fast end-to-end training from scratch. Details regarding our experience in using this model architecture are discussed in Section 5.

The key contributions of this paper are the following:

- Development of the first open-source TTS model for Macedonian language.
- Development of the first high-quality speech dataset for Macedonian language required for training TTS model, which consists of 10,433 short audio clips of a single female speaker.
- Retraining an architecture of fully end-to-end TTS synthesis, from text records to inference for Macedonian language.
- The development of a speech recorder tool for recording and matching the audio clips to the corresponding text transcriptions, suitable for easier creation of any language dataset needed for TTS model training.
- Establishment of guidelines for other researchers via discussion of the experience using state-of-the-art Deep learning networks architectures for TTS modelling.

The rest of the paper is organized, as follows. Section 2 presents the overall eminent work done in relation to creating Macedonian TTS system over the past 30 years. Section 3 describes the created dataset, data preprocessing, as well as the Deep Learning approaches experimented within the paper. Section 4 provides the results from the testing.

Section 5 presents some valuable lessons learnt when trying to achieve the desirable human-like Macedonian TTS, by exploring many TTS models implementations. We believe that those leads might be very useful for other researchers and they are not usually found in the related papers. Section 6 presents the overall conclusions that are derived from the development of the TTS system. In addition, the future directions for the possibilities to integrate the Macedonian TTS system in almost any kind of e-services in North Macedonia are also presented in Section 6.

2. Related Work

This section encompasses all of the previous prominent work done in relation to the effort to create a human-like text-to-speech synthesizer for Macedonian language. The first researches on the subject are since 1996. The authors [27] present an experimental setup for real-time TTS conversion using the classical Neural Networks approach, which was popular at that time. The results showed to be promising and ready to be integrated in a system that aimed to support humans with damaged sight that was an ongoing project since 1993; however, we did not find any paper that later describes the finished system. Next year, in 1997, the authors [28] present another effort to create a speech synthesizer that is based on time domain syllable concatenation.

In [29], the researchers present their experimental TTS system for Serbian, Croation, and Macedonian language. They characterize Serbian and Croatian to be tonal languages, meaning they have high-low pitch patterns that are permanently associated with words, whereas Macedonian language is a pitch-accented language with antepenultimate stress on most words, excluding clitics, words of foreign origin, as well as some other word groups. However, they provided a uniform dictionary-based strategy for lexical stress assignment to all three languages. The lexical stress assignment algorithm used has led to low errors for Serbian language; however, since they do not use separate speech databases for Macedonian, the speech quality has decreased, but it is reported to be still acceptable due to the fundamental similarity between phonetic inventories of the two languages.

Gerazov has done most of the valuable work in this field. In [30], the first steps towards building a model for Macedonian intonation are presented. The purpose of the models is to significantly improve the intonation generation module of their TTS system in the development phase. Further intonation analysis have been undertaken in [31], where the focus of the analysis has been set on the average intonation curves that are related to intonation phrases connected with punctuation: declaration starts, intermediates and ends, questions and exclamations. The results led to important exceptions to the existing rules and found new consistent patterns. In continuation to the research, the dynamics and variance of the intonation at different speakers has been investigated in [32]. In [33], the authors present details of the recording and segmentation process used in the creation of the unit inventory of the system in progress—"Speak Macedonian". The system itself uses a mixed-rank inventory featuring two sets of units: phones (34 basic phones, out of which for 28 there is a unique letter in the alphabet) and quasi-diphones (variation of the classic diphones—707 unique diphones extracted). The quasi-diphones is a different in the way that it encompass both the transition between the phones and the two phones together. The first paper presenting Macedonian diphone characteristics is given in [34].

Being in the final step of development of their TTS system, the authors in [35] focus on the Macedonian prosody as one of the most significant elements of the spoken language. The prosody comprises intonation, dynamics, and rhythm. Intonation, as understood from the papers presented, is essential for synthesizing speech with high-quality prosody. The authors consider two important steps in generating intonation patterns, which is, intonation prediction and pitch curve generation. Intonation prediction is the task of assigning intonation labels to the input text in accordance to the chosen intonation model. Pitch curve generation is closely related to the chosen prosody model (there are many available in the literature [36–38]). Therefore, instead of using the available models, the authors create a prosody model by themselves and present a pitch curve generation algorithm. Later, their research is focused on the emotion recognition in Macedonian speech [39].

Another trial has been done to create Macedonian TTS using concatenative speech synthesizers [40]. Those systems are considered to be simpler, since they do not rely on phonetic transitions and co-articulation, or, any other rules that are defined by linguists. Instead, concatenative speech synthesizers require well-defined recordings of speech in order to extract convenient speech segments. However, their solution mostly relies on diphones and they include certain disyllables that appear very often in the Macedonian language itself.

Some interesting experimental synthesizer is also presented in [41], where the authors follow a cross-language development of speech recognition and synthesis applications for a Macedonian language by using the bootstrapping approach from a German acoustic model. To map between the source and target languages, the authors used both knowledge-based and data-driven approaches. A new separate acoustic model was trained for HMM based synthesis. Even though the author faced low quality of the available recordings and sub-optimal phoneme mapping, the qualitative analysis showed that HMM synthesis produces satisfactory and, to some point, intelligible synthetic speech.

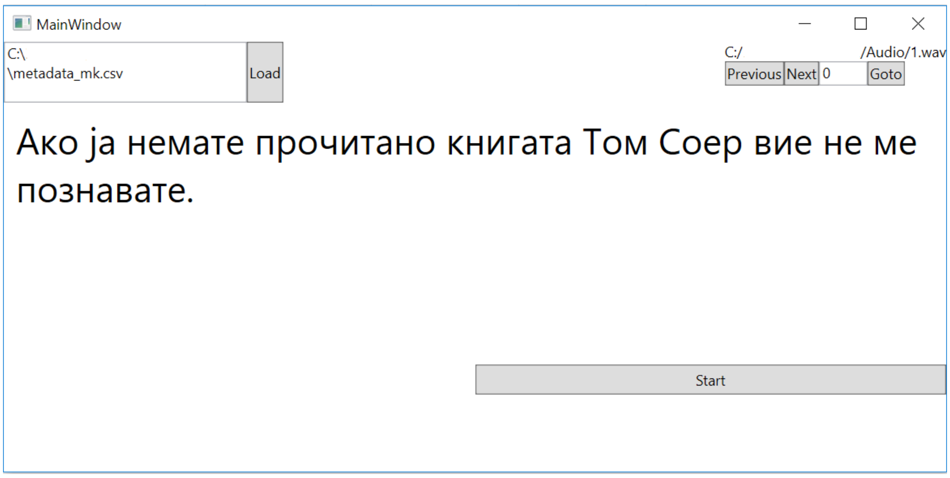
In [42], the authors try to generate a speech synthesis module for the Macedonian language based on the Russian language model. They achieved phones mapping between the Russian and Macedonian language, which showed good performance for some phones, but not so promising for some phones that do not exist in the Russian language.

A machine learning approach to morphological analysis and synthesis of Macedonian nouns is presented in [43]. Orwell's "1984" has been used for extracting a whole set of Macedonian nouns, which are then used for training a model and testing by 10-fold cross-validation. The results have been compared with 275,000 Macedonian noun forms and the accuracy achieved was above 90%.

3. Methods and Methodology

3.1. Dataset Creation

Creating appropriate and suitable dataset is essential for any intelligent system creation, and so it is in this case. We followed the guidelines of the LJ Speech Dataset [44], which is considered to be a golden standard for testing the accuracy of the DNN-based synthesizers for English language, in order to create the dataset. The dataset was created by a female speaker, using a professional microphone that will eliminate most of the noise in the background. We created a software tool that automatically serves the speaker with the consecutive samples from the text corpus and enables on-click creation of audio files in order to facilitate the recording process. The software interface is presented in the following Figure 1.



C:\
\metadata_mk.csv

C:/ /Audio/1.wav

0

Ако ја немате прочитано книгата Том Соер вие не ме познавате.

Figure 1. Software module for dataset creation.

As an input, the speaker needs to provide a .csv file that contains the whole text corpus in the format (wav_path | text_sample). Subsequently, the software automatically reads line by line and the speaker generates .wav files by clicking on the "start" button. In order to be more user-friendly and less demanding for speaker interaction, as soon as the recording process starts, the button becomes red, and the button label is changed to "stop". One more click on the button, or just clicking the "enter" key from the keyboard, and the audio file is saved in .wav format. Consequently, a new portion of sample is displayed to be read. At the top right corner, the speaker is able to manually switch among the text samples if there is a need to record some sample all over again.

For this system, the text corpus was created by reading the Macedonian translation of Mark Twain's "Huckleberry Finn", which produced 15 h of recorded material that corresponded to a total of 7686 recorded .wav files. Subsequently, the corpus was extended to 10,433 records by carefully choosing specific sentences that cover range of complex words.

The translated sentence that is displayed in Figure 1 is the first sentence from the book whose original form is "You don't know about me, without you have read a book by the name of The Adventures of Tom Sawyer, but that ain't no matter.".

3.2. Corpus Preprocessing

Before the dataset is input into the deep-learning models, several preprocessing steps are applied in order to improve the quality of the audio and text files.

The editing process of textual files required additional changes in the original texts in order to facilitate the alignment between audio files and texts. First of all, the abbreviations were replaced with their original meanings. Subsequently, the typo errors were fixed to prevent confusing the deep-learning model, making sure that high-quality text-sound corpus is achieved. Hereupon, the replacement of numerical values is performed (like ordinal numbers, years, quantitative values, etc.) with their textual representations. These transformations were also applied in inference phase, because the models would not be aware of the language rules that are used in Macedonian language.

Audio files require additional preprocessing in order to remove the noise that may appear and cease the training process. Therefore, the silence at the beginning was trimmed as well as at the end of the audio samples in the dataset. It proved that trimming facilitates the alignment between the text utterances and audio samples, which decreases the time requirements for training. Next, the large sentences were split into smaller sentences in order to lower the model's processing requirements, only allowing recorded sentences up to 10 s long. Consequently, it helped to use larger batch size when using GPU with lower performances.

3.3. Deep Learning Approach

Deep Voice 3 is chosen to be the most appropriate for Macedonian language TTS system creation, since it outperforms other models in terms of the trade-off between speed and generated speech quality according to our experimental results (more details are found in Section 5). It is able to synthesize more than 10M sentences per day [23] by using the leverage of the GPU. Even more, its sound-to-phoneme mapping ability is the most suitable for Macedonian language, which is consistent and phonemic in practice, and it follows the principle one grapheme per phoneme. This one-to-one correspondence is described by the principle, "write as you speak and read as it is written" [45].

Deep Voice 3 is a fully convolutional architecture for speech synthesis. Its character-to-spectrogram architecture enables fully parallel computation and the training is much faster than at the RNN architectures. The quality of the architecture has been evaluated on many datasets that encompass 820 h of audio data from 2484 speakers [46]. The architecture generates monotonic attention behavior, which avoids error modes that are common at sequence-to-sequence models.

Deep Voice 3 architecture converts the characters, phonemes, stresses, and other textual features into a variety of vocoder parameters that are used as inputs into audio waveform synthesis models. Mel-band spectrograms, linear-scale log magnitude spectrograms, fundamental frequency, spectral envelope, and aperiodicity parameters are all vocoder parameters.

Mainly, the architecture consists of an encoder, decoder, and converter layer. The encoder layer transforms the previously defined textual features into internally learnt features representations. Those features are in a (key, value) form and they are fed into the attention-based decoder. The decoder uses its convolutional attention mechanism to transform those features into low-dimensional audio representation, i.e., mel-scale log magnitude spectrograms that correspond to the output audio. The hidden layers of the decoder are fed into the third converter layer, which is capable of predicting the acoustic features for waveform synthesis. Figure 2 presents the detailed architecture of the model and the methodology workflow.

In the following Figure 2, the whole workflow is depicted via four separate stages. The first is the Dataset creation showing the sources, the process of recording the sentences, as well as the production of files in a suitable format that is recognized by the TTS model. The following stage is Text and audio preprocessing and it is responsible for assuring high-quality input into the TTS model. It comprises four preprocessing steps that refer to the audio files and five preprocessing steps that refer to the corresponding text files. The audio files underwent noise removal in order to remove the background noise as a result from the technical equipment used for recording; then, silence trimming to equalize the beginning and end of the records; amplification to strengthen the spoken speech, and long audio segmentation into shorter parts of maximum 10 s (which was found to be the case in Ljspeech dataset [44]). The corresponding text files were improved in a way that all typo errors were fixed; also the unrecognized symbols and unnecessary white spaces were removed; the numbers, ordinals, and units were converted into full written words as well as the abbreviations, and eventually, the long sentences were split to match the audio files that were segmented in the previous step.

As soon as the dataset is ready, it is input in the TTS model whose architecture is previously explained in the same section. The training of the model is followed by validation, which means that each 10K steps are evaluated before the training process proceeds. The evaluation is done on external, unknown sentences that provide insight into the advancement of the learnt dependencies between the dataset and the hidden layer weights. As can be seen, the TTS model stage is followed by the Inference stage. This stage itself is independent from the previous stage and it uses the leverage of the already created checkpoints. As new text is input, it is preprocessed in the same manner, as explained in the Text and audio preprocessing stage, and it is is additionally segmented. This segmentation is done in order to prevent inferences with duration over 10 s. As the input sentences are segmented, they are batched with the aim to preserve the order. Each of them is predicted by the checkpoint upon which an intelligible speech is synthesized. The segments are put together during the audio postprocessing and concatenation step, and they are presented as a single .wav file.

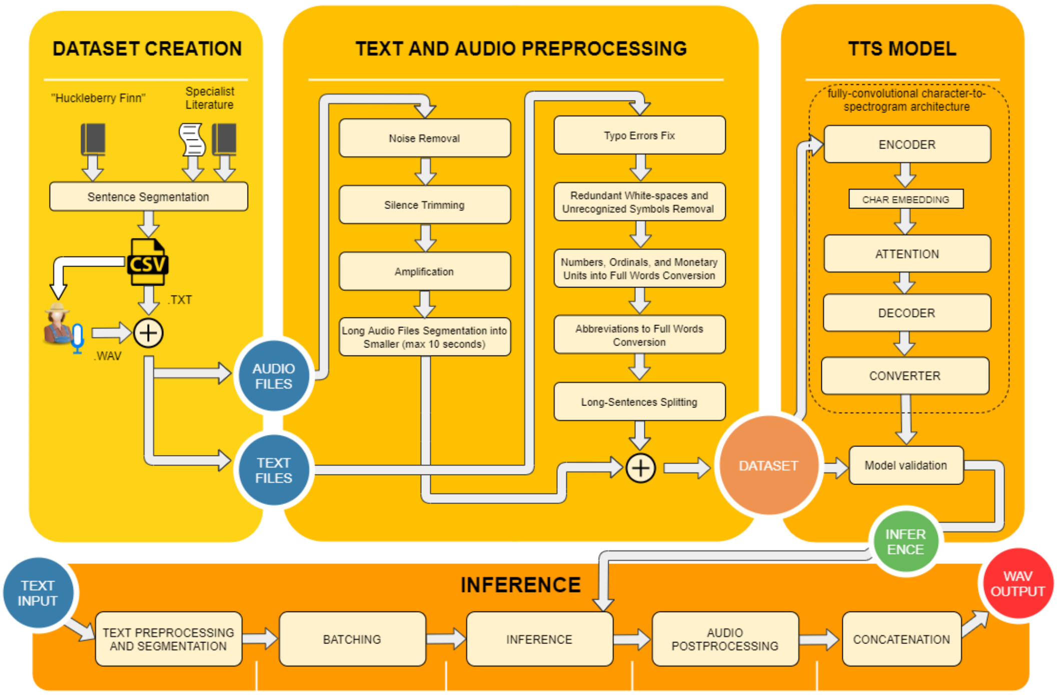


Figure 2. MAKEDONKA's Text-to-Speech methodology (training and inference phases).

Appropriate settings of the model's hyperparameters is crucial for producing a human-like TTS system. All of the related implementations are using the Ljspeech dataset [44], meaning that the settings are for the English alphabet and the corresponding set of phonetic transcription codes—the ARPAbet. Even though the official Macedonian alphabet is Cyrillic, we decided to use transliteration cleaners in order to represent it via the English alphabet. The ARPAbet was completely omitted, since it is not useful for Macedonian language. Omitting the ARPAbet, required changing the pronunciation probability parameter, which is highly related to retrieving the phonemes representations of the words from the phonemes dictionary. If not set appropriately, the outcome is incomplete speech with shortened words and sentences. Table 1 provides the rest of the hyperparameters. Deep Voice 3 suggested that parameters worked perfectly for our model, thus we used the same hyperparameters without increasing the demand due to our resource limitations.

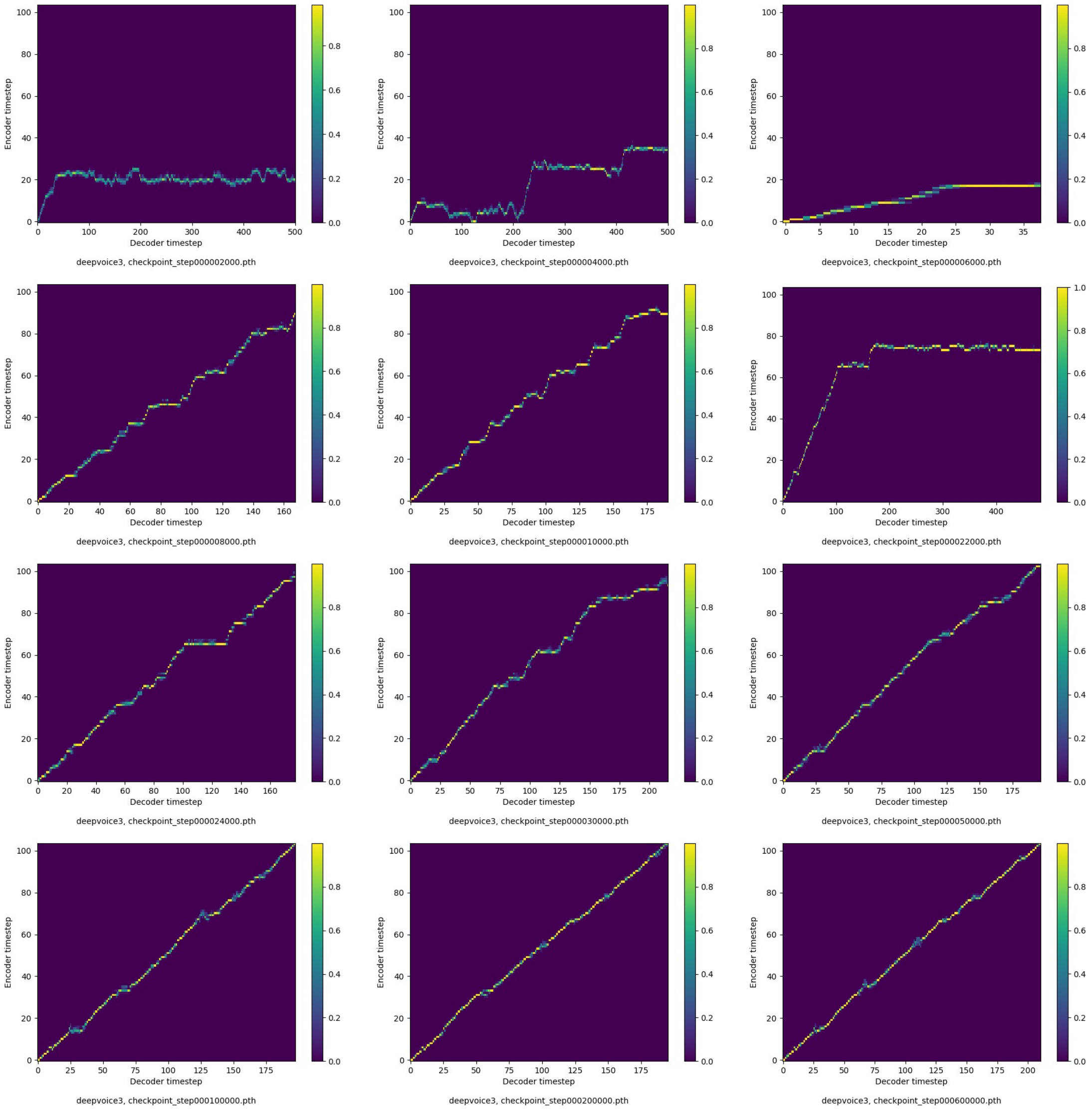
Table 1. Model's hyperparameters settings.

Hyperparameters	
"replace_pronunciation_prob": 0.0,	"converter_channels": 256,
"speaker_embed_dim": 16,	"query_position_rate": 1.0,
"num_mels": 80,	"key_position_rate": 1.385,
"fmin": 125,	"num_workers": 2,
"fmax": 7600,	"masked_loss_weight": 0.5,
"fft_size": 1024,	"priority_freq": 3000,
"hop_size": 256,	"priority_freq_weight": 0.0,
"sample_rate": 22050,	"binary_divergence_weight": 0.1,
"preemphasis": 0.97,	"guided_attention_sigma": 0.2,
"min_level_db": −100,	"batch_size": 32,
"ref_level_db": 20,	"adam_beta1": 0.5,
"rescaling_max": 0.999,	"adam_beta2": 0.9,
"downsample_step": 4,	"adam_eps": 0.000006,
"outputs_per_step": 1,	"initial_learning_rate": 0.0005,
"embedding_weight_std": 0.1,	"nepochs": 2000,
"speaker_embedding_weight_std": 0.01,	"weight_decay": 0.0,
"padding_idx": 0,	"clip_thresh": 0.1,
"max_positions": 512,	"checkpoint_interval": 1000,
"dropout": 0.050000000000000044,	"eval_interval": 10000,
"kernel_size": 3,	"window_ahead": 3,
"text_embed_dim": 256,	"window_backward": 1,
"encoder_channels": 512,	"power": 1.4
"decoder_channels": 256,	

4. Results

To achieve single-speaker synthesis, approximately 20 h of Macedonian high-quality speech audio dataset recorded at a sample rate of 22.5 kHz was used. The training was performed by using NVIDIA Tesla P100 with 16GB RAM. The total training time by using a batch size of 16 took 21 days and 16 h. The training was completed after 620K steps. Figure 3 presents the attention changing during the training process. The model started to produce an intelligible, understandable, and partially human-like speech after 50 K steps, as observed from the figure. Hereafter, the model started to improve itself by loosing the "robotic" component in the synthesized speech and achieved completely human-like speech until the end of the training.

At each checkpoint, the model was evaluated on seven different sentences that are carefully chosen to be specific in order to test the ability of the models. By the term specific, it means that the examples cover special cases in the Macedonian language, such as: long words; compound words; comma somewhere in the sentence to check whether the model makes the appropriate pause when synthesizes the speech; sentences ending with fullstop, question mark, and exclamation mark to check whether the model is able to change the intonation in the synthesized speech; and, tongue twisters, and tongue twisters containing words with multiple adjacent consonants, such as the word "Shtrk". The audio files from the synthesized speech across the checkpoints are available on GitHub https://f-data.github.io/TTS/.



deepvoice3, checkpoint_step000002000.pth

deepvoice3, checkpoint_step000004000.pth

deepvoice3, checkpoint_step000006000.pth

deepvoice3, checkpoint_step000008000.pth

deepvoice3, checkpoint_step000010000.pth

deepvoice3, checkpoint_step000022000.pth

deepvoice3, checkpoint_step000024000.pth

deepvoice3, checkpoint_step000030000.pth

deepvoice3, checkpoint_step000050000.pth

deepvoice3, checkpoint_step000100000.pth

deepvoice3, checkpoint_step000200000.pth

deepvoice3, checkpoint_step000600000.pth

Figure 3. Attention changing during training process.

Figure 4 presents the metrics that speak of the performance of the training models. Loss function is a metric that refers to the accuracy of the prediction. The main objective is to minimize the model errors or minimize the loss function. Thus, the Deep learning algorithm will repeat as many times needed in order the loss to reach as flatter line shape as possible. In our case, the loss functions behaves in a desired manner, it gradually decreases, converging to a value of 0.1731 after 162.2 K steps in four days and 11 h of training.

Learning rate plays a vital role in minimizing the loss function. It dictates the speed at which we want our model to learn. This value must be set properly, since if not, for example, setting it too high, the model will not have time to learn anything and, thus, the results will be poor. The initial learning rate was set to 0.0005, as shown in Table 1. After four days and 5 h of training, or 151.2 K steps, it decreases to a value of 0.000081335.

The gradient norm that is presented in the same figure calculates the L2 norm of the gradients of the last layer of the Deep learning network. It is an indicator showing whether the weights of the Deep learning network are properly updated. If its value is too small, it might indicate vanishing gradient. This problem affects the upper layers of the Deep learning network, making it really hard for the network to learn and tune the parameters. On the contrary, if its value is too high, it may indicate exploding a gradient phenomenon. In such case, the model is unstable and it is not able to learn from data, since the accumulation of large error gradients during the training process result in very large updates in the Deep learning model weights.

The last performance metric refers to the ability of the model to predict the mel-spectrograms. The L1 norm of the metric is decreasing across the iteration steps, reaching 0.03304 after 172.8 K steps, or four days and 18 h of training.

The model and the samples can be found on GitHub https://github.com/f-data/TTS-Makedonka.

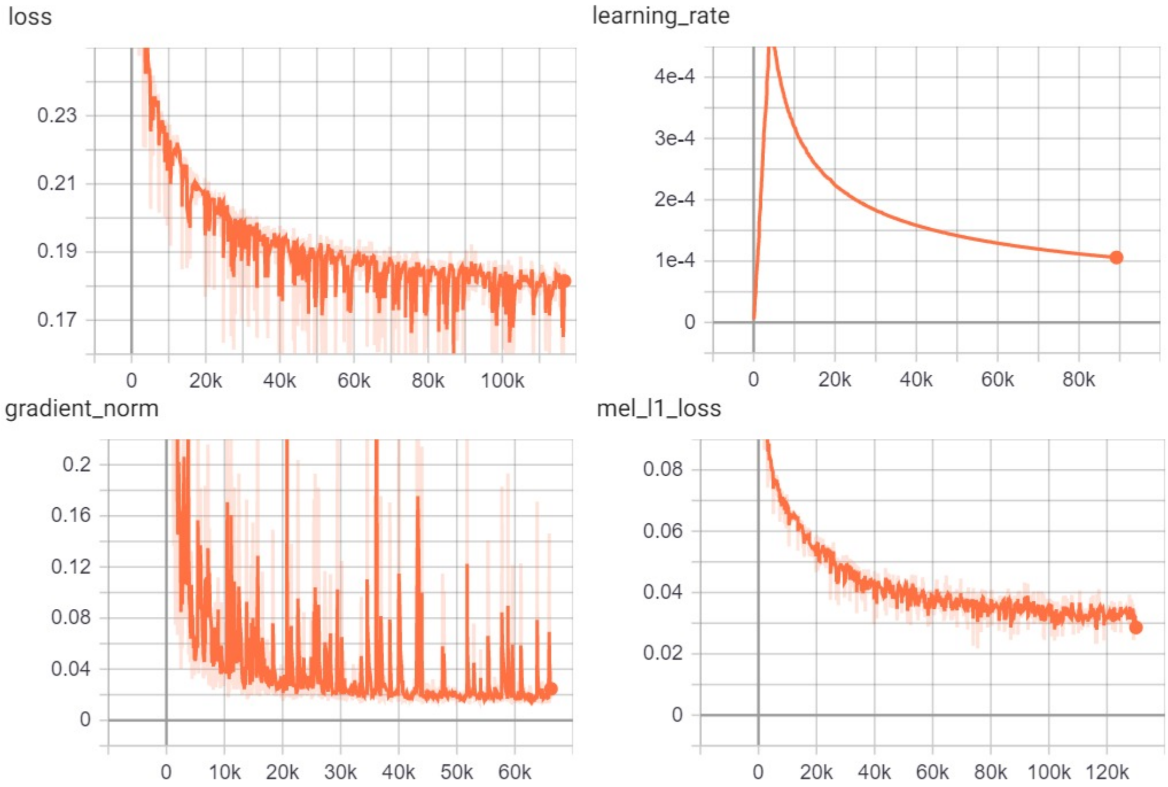


Figure 4. Tensorboard scalars of the learning models.

The quality of the model has been assessed on the basis of reliable and valid listening tests in order to assess overall TTS model performance—the Mean Opinion Score (MOS) [47,48].

Table 2 presents the obtained MOS values for the ground truth and the selected TTS model, which are 4.6234 ± 0.2739 and 3.9285 ± 0.1210, correspondingly. Fifteen distinct listeners performed the assessment on 40 original (ground truth) and 40 synthesized audio files.

Table 2. Mean Opinion Score (MOS) Results.

Experiment	Subjective five-Scale MOS
Ground Truth	4.6234 ± 0.2739
Deep Voice 3	3.9285 ± 0.1210

According to [47,48], the obtained value indicates good quality audio, with no additional effort to understand the words, distinguishable sounds, not annoying pronunciation, preferred speed, and pleasant voice.

5. Discussion

During the process of selection, the most appropriate model for the creation of Macedonian TTS system, many state-of-the-art models were experimentally evaluated besides Deep Voice 3, such as: multiple implementations of Tacotron2 (both Tensorflow [49] and Pytorch [50]), implementation of

Tacotron 2 for Russian language [51], Fastspeech [49], FastSpeech2 [49], ClariNet [25], and also audio synthesizers, such as Melgan [49] and Waveglow [52]. Melgan and Waveglow both showed very good performance, even after 100K iterations. Additionally, pretrained Melgan and Waveglow models in English language could be successfully used as audio synthesizers for Macedonian language without any difficulties.

Tacotron 2 is the most famous and promising model to produce a human-like speech. However, the training process takes days before intelligible speech is synthesized. We also considered a distillated version used to create TTS for Russian language. By principles of transfer learning ,we tried to fine-tune the Russian TTS model; however, the experiments were not as successful as expected.

FastSpeech and FastSpeech2 are much faster that Tacotron 2, however, they are not completely independent. Actually, they rely on models for duration prediction that could be either Tacotron 2 model or MFA. Training TTS systems from scratch for language other than English requires lots of time to prepare for using FastSpeech and FastSpeech2.

ClariNet's architecture is an extension of DeepVoice 3 and, therefore, it was taken into consideration for training our TTS model. However, ClariNet requires a pre-trained Wavenet model, meaning that we need to train reliable Wavenet model from scratch for Macedonian language and then to proceed training a ClariNet model. We were unable to accomplish this in a reasonable time limit due to our resource limitations and thus, we chose to work with Deep Voice 3.

6. Conclusions and Future Work

This paper presents a successful effort to train a human-like TTS model for the Macedonian language. After many attempts to implement an efficient and humanoid TTS system for the Macedonian language in the last 30 years, we are the first who built it and published the model that is available to use as a module in any kind of software that needs its service.

The methodology presented in the paper relies on a previously confirmed Deep learning-based methodology—Deep Voice 3. We built software for new records management and created approximately 20 h-long training corpus from scratch in order to achieve a high-quality model. The dataset has been preprocessed by following the example of the Ljspeech dataset [44], which is considered to be the golden standard for training English language TTS systems. The deep neural network has been adjusted according to the Macedonian language needs.

Intelligible speech has been synthesized after 56 K steps of training, and acceptable quality has been achieved, even after 100 K steps of training. However, the model has been improved in the later steps, and the robotic-like components in the synthesized speech have been almost removed after 200 K steps of training. The quality of the generated audio files has been assessed while using the MOS metric, which is commonly used to assess the TTS systems.

Many intelligent systems may benefit from such a TTS system, as are the recommendation systems developed for the social media networks [53–55], by establishing an interaction with the users and, thus, improving their experience by achieving human-like communication.

In future work, we will extend the dataset with new sentences that cover as many different fields as possible, including many new words with different pronunciation complexity. We will also work on improving the pronunciation of words with irregular accents in the Macedonian language. The word stress in the Macedonian language is antepenultimate and dynamic, which means that it falls on the third from last syllable in words with three or more syllables, and on the first or only syllable in other words. However, this rule is sometimes disregarded when the word is among the more recent ones in the language or it is from a foreign language.

The training dataset, the synthesized samples at different checkpoints, the source code, and the trained model are publicly available for further improvement on GitHub https://github.com/f-data/TTS-Makedonka.

Author Contributions: Conceptualization, K.M., A.K.R., D.T., T.E. and M.S.; Formal analysis, T.E.; Funding acquisition, T.E.; Investigation, D.T. and M.S.; Methodology, K.M., T.E. and M.S.; Project administration, A.K.R.; Resources, M.S.; Software, K.M.; Supervision, M.S.; Validation, K.M. and D.T.; Visualization, K.M.; Writing—original draft, K.M., T.E. and M.S.; Writing—review & editing, K.M., A.K.R., T.E. and M.S. All authors have equally contributed to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by project from the Slovenian Research Agency [research core grant number P2-0098], and was partially funded by Faculty of Computer Science and Engineering.

Acknowledgments: The work was inspired by Erasmus+ KA203 FAST: Fostering Accessible Study Technologies—Accessible Learning Management System in Humanities and Social Sciences (project reference: 2018-1-MK01-KA203-047104).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ning, Y.; He, S.; Wu, Z.; Xing, C.; Zhang, L.J. A review of deep learning based speech synthesis. *Appl. Sci.* **2019**, *9*, 4050. [CrossRef]
2. Murray, I.R. Simulating Emotion in Synthetic Speech. Ph.D. Thesis, University of Dundee, Dundee, UK, 1989.
3. Tokuda, K.; Yoshimura, T.; Masuko, T.; Kobayashi, T.; Kitamura, T. Speech parameter generation algorithms for HMM-based speech synthesis. In Proceedings of the 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing (Cat. No. 00CH37100), Istanbul, Turkey, 5–9 June 2000; Volume 3, pp. 1315–1318.
4. Yoshimura, T.; Tokuda, K.; Masuko, T.; Kobayashi, T.; Kitamura, T. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In Proceedings of the Sixth European Conference on Speech Communication and Technology, Budapest, Hungary, 5–9 September 1999.
5. Yamagishi, J.; Nose, T.; Zen, H.; Ling, Z.H.; Toda, T.; Tokuda, K.; King, S.; Renals, S. Robust speaker-adaptive HMM-based text-to-speech synthesis. *IEEE Trans. Audio Speech Lang. Process.* **2009**, *17*, 1208–1230. [CrossRef]
6. Ratnaparkhi, A. *A Simple Introduction to Maximum Entropy Models for Natural Language Processing*; IRCS Technical Reports Series; University of Pennsylvania: Philadelphia, PA, USA, 1997; p. 81.
7. Gu, L.; Gao, Y.; Liu, F.H.; Picheny, M. Concept-based speech-to-speech translation using maximum entropy models for statistical natural concept generation. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 377–392. [CrossRef]
8. Sridhar, V.K.R.; Bangalore, S.; Narayanan, S.S. Exploiting acoustic and syntactic features for automatic prosody labeling in a maximum entropy framework. *IEEE Trans. Audio Speech Lang. Process.* **2008**, *16*, 797–811. [CrossRef] [PubMed]
9. Coorman, G.; Deprez, F.; De Bock, M.; Fackrell, J.; Leys, S.; Rutten, P.; DeMoortel, J.; Schenk, A.; Van Coile, B. Speech Synthesis Using Concatenation of Speech Waveforms. U.S. Patent 6,665,641, 12 January 2003.
10. Moulines, E.; Charpentier, F. Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech Commun.* **1990**, *9*, 453–467. [CrossRef]
11. Charpentier, F.; Stella, M. Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal (ICASSP'86), Tokyo, Japan, 7–11 April 1986; Volume 11, pp. 2015–2018.
12. Zen, H.; Tokuda, K.; Black, A.W. Statistical parametric speech synthesis. *Speech Commun.* **2009**, *51*, 1039–1064. [CrossRef]
13. Ze, H.; Senior, A.; Schuster, M. Statistical parametric speech synthesis using deep neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–30 May 2013; pp. 7962–7966.
14. Zen, H.; Senior, A. Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 3844–3848.
15. Lu, H.; King, S.; Watts, O. Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. In Proceedings of the Eighth ISCA Workshop on Speech Synthesis, Barcelona, Spain, 31 August–2 September 2013.
16. Wang, Y.; Skerry-Ryan, R.; Stanton, D.; Wu, Y.; Weiss, R.J.; Jaitly, N.; Yang, Z.; Xiao, Y.; Chen, Z.; Bengio, S.; et al. Tacotron: Towards end-to-end speech synthesis. *arXiv* **2017**, arXiv:1703.10135.

17. Shen, J.; Pang, R.; Weiss, R.J.; Schuster, M.; Jaitly, N.; Yang, Z.; Chen, Z.; Zhang, Y.; Wang, Y.; Skerrv-Ryan, R.; et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 5–20 April 2018; pp. 4779–4783.
18. Yasuda, Y.; Wang, X.; Takaki, S.; Yamagishi, J. Investigation of enhanced Tacotron text-to-speech synthesis systems with self-attention for pitch accent language. In Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019), Brighton, UK, 12–17 May 2019; pp. 6905–6909.
19. Ren, Y.; Ruan, Y.; Tan, X.; Qin, T.; Zhao, S.; Zhao, Z.; Liu, T.Y. Fastspeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 3171–3180.
20. Ren, Y.; Hu, C.; Qin, T.; Zhao, S.; Zhao, Z.; Liu, T.Y. FastSpeech 2: Fast and High-Quality End-to-End Text-to-Speech. *arXiv* **2020**, arXiv:2006.04558.
21. Arik, S.Ö.; Chrzanowski, M.; Coates, A.; Diamos, G.; Gibiansky, A.; Kang, Y.; Li, X.; Miller, J.; Ng, A.; Raiman, J.; et al. Deep voice: Real-time neural text-to-speech. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 195–204.
22. Gibiansky, A.; Arik, S.; Diamos, G.; Miller, J.; Peng, K.; Ping, W.; Raiman, J.; Zhou, Y. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 2962–2970.
23. Ping, W.; Peng, K.; Gibiansky, A.; Arik, S.O.; Kannan, A.; Narang, S.; Raiman, J.; Miller, J. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. *arXiv* **2017**, arXiv:1710.07654.
24. Tachibana, H.; Uenoyama, K.; Aihara, S. Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 4784–4788.
25. Ping, W.; Peng, K.; Chen, J. Clarinet: Parallel wave generation in end-to-end text-to-speech. *arXiv* **2018**, arXiv:1807.07281.
26. Oord, A.V.D.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; Kavukcuoglu, K.; Driessche, G.V.D.; Lockhart, E.; Cobo, L.C.; Stimberg, F.; et al. Parallel wavenet: Fast high-fidelity speech synthesis. *arXiv* **2017**, arXiv:1711.10433.
27. Josifovski, L.; Mihajlov, D.; Djordjevik, D. Text-to-Speech Conversion for Macedonian as Part of a System for Support of Humans with Damaged Sight. In Proceedings of the 18th International Conference on Information Technology Interfaces (ITI), Pula, Croatia, 18–21 June 1996; Volume 96, pp. 61–66.
28. Josifovski, L.; Mihajlov, D.; Gorgevik, D. Speech synthesizer based on time domain syllable concatenation. In Proceedings of the SPECOM, Cluj-Napoca, Romania, 27–30 October 1997; Volume 97, pp. 165–170.
29. Delić, V.; Sečujski, M.; Pekar, D.; Jakovljević, N.; Mišković, D. A Review of AlfaNum Speech Technologies for Serbian, Croatian and Macedonian. In Proceedings of the International Language Technologies Conference IS-LTC, Ljubljana, Slovenia, 9–10 October 2006; Volume 6, pp. 257–260.
30. Gerazov, B.; Ivanovski, Z.; Bilibajkic, R. Modeling Macedonian intonation for text-to-speech synthesis. In Proceedings of the DOGS 2010, Iriski Venac, Serbia, 16–18 December 2010; pp. 16–18.
31. Gerazov, B.; Ivanovski, Z. Analysis of intonation in the Macedonian language for the purpose of text-to-speech synthesis. In Proceedings of the EAA EUROREGIO 2010, Ljubljana, Slovenia, 15–18 September 2010.
32. Gerazov, B.; Ivanovski, Z. Analysis of intonation dynamics in Macedonian for the purpose of text to speech synthesis. In Proceedings of the TELFOR 2010, Belgrade, Serbia, 23–25 November 2010.
33. Gerazov, B.; Ivanovski, Z. The Construction of a Mixed Unit Inventory for Macedonian Text-to-Speech Synthesis. In Proceedings of the International Scientific-Professional Symposium INFOTEH, Jahorina, Bosnia and Herzegovina, 16–18 March 2011.
34. Gerazov, B.; Ivanovski, Z. Diphone Analysis of the Macedonian Language for the Purpose of Text-to-Speech Synthesis. In Proceedings of the ICEST 2009, Veliko Tarnovo, Bulgaria, 25–27 June 2009.
35. Gerazov, B.; Ivanovski, Z. Generation of pitch curves for Macedonian text-to-speech synthesis. In Proceedings of the 6th Forum Acusticum, Aalborg, Denmark, 27 June–1 July 2011.
36. Kochanski, G.; Shih, C. Prosody modeling with soft templates. *Speech Commun.* **2003**, *39*, 311–352. [CrossRef]
37. Taylor, P. *Text-To-Speech Synthesis*; Cambridge University Press: Cambridge, UK, 2009.

38. Taylor, P. Analysis and synthesis of intonation using the tilt model. *J. Acoust. Soc. Am.* **2000**, *107*, 1697–1714. [CrossRef] [PubMed]
39. Gerazov, B.; Peev, G.; Hristov, M.; Ivanovski, Z. Towards speech emotion recognition in Macedonian. In Proceedings of the ETAI 2015, Ohrid, North Macedonia, 24–26 September 2015.
40. Chungurski, S.; Kraljevski, I.; Mihajlov, D.; Arsenovski, S. Concatenative speech synthesizers and speech corpus for Macedonian language. In Proceedings of the 30th International Conference on Information Technology Interfaces (ITI 2008), Dubrovnik, Croatia, 23–26 June 2008; pp. 669–674.
41. Kraljevski, I.; Strecha, G.; Wolff, M.; Jokisch, O.; Chungurski, S.; Hoffmann, R. Cross-language acoustic modeling for Macedonian speech technology applications. In Proceedings of the International Conference on ICT Innovations 2012; Ohrid, North Macedonia, 13–15 September 2012; pp. 35–45.
42. Mingov, R.; Zdravevski, E.; Lameski, P. Application of russian language phonemics to generate macedonian speech recognition model using sphinx. In Proceedings of the ICT Innovations 2016, Ohrid, North Macedonia, 5–7 September 2016.
43. Ivanovska, A.; Zdravkova, K.; Erjavec, T.; Džeroski, S. Learning rules for morphological analysis and synthesis of Macedonian nouns, adjectives and verbs. In Proceedings of the 5th Slovenian and 1st International Language Technologies Conference, Ljubljana, Slovenia, 9–13 October 2006; pp. 140–145.
44. Ito, K. The lj Speech Dataset. 2017. Available online: https://keithito.com/LJ-Speech-Dataset/ (accessed on 26 September 2020).
45. Friedman, V. Macedonian, Slavic and Eurasian Language Resource Center (SEELRC). 2001. Available online: shorturl.at/blAC0 (accessed on 26 September 2020).
46. Ping, W.; Peng, K.; Gibiansky, A.; Arik, S.O.; Kannan, A.; Narang, S.; Raiman, J.; Miller, J. Deep voice 3: 2000-speaker neural text-to-speech. In Proceedings of the ICLR, Vancouver, Canada, 30 April–3 May 2018; pp. 214–217.
47. Viswanathan, M.; Viswanathan, M. Measuring speech quality for text-to-speech systems: Development and assessment of a modified mean opinion score (MOS) scale. *Comput. Speech Lang.* **2005**, *19*, 55–83. [CrossRef]
48. Ribeiro, F.; Florêncio, D.; Zhang, C.; Seltzer, M. Crowdmos: An approach for crowdsourcing mean opinion score studies. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 2416–2419.
49. Anh, M. Tensorflow TTS. Available online: https://github.com/TensorSpeech/TensorFlowTTS (accessed on 26 September 2020).
50. Didur, I. Tacotron 2. Available online: https://github.com/ide8/tacotron2 (accessed on 26 September 2020).
51. Shmyrev, N.V. Implementation of Tacotron 2 for Russian language. Available online: https://github.com/alphacep/tn2-wg (accessed on 26 September 2020).
52. NVIDIA. Waveglow. Available online: https://github.com/NVIDIA/waveglow (accessed on 26 September 2020).
53. Amato, F.; Moscato, V.; Picariello, A.; Sperlí, G. Kira: A system for knowledge-based access to multimedia art collections. In Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 30 January–1 February 2017; pp. 338–343.
54. Amato, F.; Moscato, V.; Picariello, A.; Sperlí, G. Recommendation in social media networks. In Proceedings of the 2017 IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, CA, USA, 19–21 April 2017; pp. 213–216.
55. Amato, F.; Castiglione, A.; Moscato, V.; Picariello, A.; Sperlì, G. Multimedia summarization using social media content. *Multimed. Tools Appl.* **2018**, *77*, 17803–17827. [CrossRef]