Karlsruhe Institute of Technology

A Text-to-Speech system based on Deep Neural Networks

Bachelor's Thesis
of

Arsenii Dunaev

KIT Department of Informatics
Institute for Anthropomatics and Robotics (IAR)
Interactive Systems Labs (ISL)

Reviewers:	Prof. Dr. Alexander Waibel Prof. Dr. Tamim Asfour
Advisors:	M.Sc. Stefan Constantin

Duration: July 12th, 2019 – November 11th, 2019

Erklärung:

Ich versichere hiermit, dass ich die Arbeit selbstständig verfasst habe, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Karlsruhe, den 11. November 2019

Arsenii Dunaev

Abstract:

Based on the previous success in the Text-to-Speech field using Deep Neural Networks (Shen et al., 2018), this work aims at training and evaluation of an attention-based sequence-to-sequence neural voice generation system. Chosen fully-convolutional architecture matches state-of-the-art techniques in speech synthesis while training an order of magnitude faster than analogous recurrent architectures (Ping et al., 2018).

This thesis also considers a great number of open-source speech datasets for both English and German languages, determining the best one and its key characteristics for training of the Text-to-Speech system.

To achieve a high quality of synthesized speech various training techniques are used and described in detail during this work. Such methods as concatenation of separate audio-files in a larger ones with uniform distribution of lengths (between 0 and 20 seconds) and trimming of silences with modern tools allow the model to produce more intelligible speech utterances of any duration as well as more natural speech utterances without clipping in the end, which is considered to be an issue for many Text-to-Speech systems. This work also addresses the training of Text-to-Speech models for other languages by specifying new frontend (e.g. for German model).

Several models are evaluated through user studies by measuring three criteria: intelligibility, comprehensibility, and naturalness. The most successful English and German models achieve the Word Error Rate of 23.48% and 14.21% respectively, and a higher than average degree of comprehensibility and naturalness. In conclusion, suggestions and recommendations on training of high-performance Text-to-Speech language models for future work are given.

Kurzzusammenfassung:

Basierend auf früheren Erfolgen im Bereich Text-to-Speech mit Deep Neural Networks (Shen et al., 2018), stellt diese Arbeit als Ziel das Training und die Evaluierung einer Attention-basierten Sequenz-zu-Sequenz neuronalen System zur Erzeugung der menschlichen Stimme. Die ausgewählte fully-convolutional Architektur entspricht dem Stand der Technik in Sprachsynthese, während das System um eine Größenordnung schneller trainiert wird als bei analogen recurrent Architekturen (Ping et al., 2018).

Diese Thesis betrachtet auch eine große Anzahl von Open-Source-Sprachdatensätzen für englische und deutsche Sprachen, um den besten Datensatz und seine wichtigsten Merkmale für das Training des Text-to-Speech Systems zu ermitteln.

Um eine hohe Qualität der synthetisierten Sprache zu erreichen, werden verschiedene Trainingstechniken in dieser Arbeit verwendet und detailliert beschrieben. Solche Methoden wie die Konkatenation separater Audiodateien in größere Dateien mit gleichmäßiger Verteilung der Tondauern (zwischen 0 und 20 Sekunden) und das Trimmen von Pausen mit modernen Werkzeugen ermöglichen es dem Modell, verständlichere Sprachausgaben von beliebiger Dauer beziehungsweise natürlichere Sprachausgaben ohne Abschneiden am Ende der Aussage zu erzeugen, was für viele Text-to-Speech Systeme ein Problem darstellt. Diese Arbeit befasst sich auch mit dem Training von Text-to-Speech Modellen für andere Sprachen, indem ein neues Frontend (z. B. für das deutsche Modell) spezifiziert wird.

Mehrere Modelle werden durch Benutzerstudien anhand von drei Kriterien bewertet: Verständlichkeit, Nachvollziehbarkeit und Natürlichkeit. Die erfolgreichsten englischen und deutschen Modelle erreichen eine Wortfehlerrate von 23,48% bzw. 14,21% und einen überdurchschnittlichen Grad an Nachvollziehbarkeit und Natürlichkeit. Im Ausblick werden abschließende Vorschläge und Empfehlungen zum Training hochperformanter Text-to-Speech Modelle gegeben.

Contents

1. Introduction

1.1. Motivation

Natural Language Processing (NLP) is one of the widest parts of the development of 'human-like' machines (such as robots) among computer vision and artificial intelligence itself. One of the largest and most sophisticated tasks of NLP is speech synthesis. Inspired by speaking machines in science fiction, scientists and engineers have been fascinating and studying the ways of establishing speech by machines for many years. Started with "acoustic-mechanical speech machines" (Wolfgang von Kempelen's speaking machine in 1769-1791 (von Kempelen, 1791)) and continued with the first computer-based speech synthesis systems in the late 1950s (Bell Labs: synthesizing speech on IBM 704 mentioned in Mullennix and Stern (2010)), the technology advanced fast with Deep Neural Networks (DNNs) in recent years.

Speaking about the speech synthesis mostly refers to the Text-to-Speech (TTS) systems: ability of the computer to read text aloud. The conventional methods (such as concatenative, source-filter or Hidden Markov Model (HMM)-based synthesis) achieved good quality in synthesizing comprehensible and natural human speech, but they still possess drawbacks and limitations (e.g. long development time, high complexity of algorithms or large storage capacity). Neural networks, already extremely popular in other artificial intelligence branches, offer a chance to escape some limitations and produce a natural 'human' voice when a huge amount of sample data is available. With the help of DNNs many nowadays companies have made a breakthrough in the development of voice assistants products like Apple's Siri, Google Assistant or Amazon Alexa, which are approaching the quality of the human voice.

Speech synthesis is still one of the significant vital assistive technology tools, which allows environmental barriers to be removed for people with a wide range of disabilities such as blindness or loss of speech.

The best modern TTS systems are usually commercial (i.e. its architecture or the dataset, on which the system is trained, are closed-source). This work will make the first steps towards training and evaluation of the TTS system, which is based on DNNs and is completely open-source.

1.2. Goal of this work

This work aims at training and evaluating of the single end-to-end multi-modal neural network that:

- produces comprehensible and natural male voice (since most of the previous open-source TTS systems are trained on the commonly known female voice dataset prepared by Ito (2017), which are usually better than analogous trained on male speech datasets);
- has a low latency of synthesizing (<2 seconds);
- has a low operating power of synthesizing the speech (use of two Central Processing Unit (CPU)-threads completely without the Graphics Processing Unit (GPU)).

The training will be conducted based on an implementation (Yamamoto et al., 2018) of the *Deep Voice 3* TTS system by *Baidu Research* (Ping et al., 2018).

To successfully train the TTS system a large natural human speech dataset have to be found. This work will cover a vast amount of freely available speech datasets and compare them to each other in order to determine:

1. the most suitable one for the given problem;
2. the optimal size of the dataset needed to suffice the high quality of synthesized speech.

This way, the quality of the speech must correspond to the high quality of modern TTS systems such as *Tacotron 2* by *Google* (Shen et al., 2018). In addition, the trained model should produce speech promptly without considerable loss in naturalness or intelligibility.

To conclude, this thesis aims at exploring the use of neural networks in modern TTS systems and possible future work in this field.

2. Fundamentals

The following chapter describes useful basics in the field of Artificial Neural Networks (ANNs), especially the relevant techniques for this work.

2.1. Perceptron

A perceptron is the smallest unit of an Artificial Neural Network. The perceptron algorithm was invented in 1958 by Frank Rosenblatt as a simplified model of a biological neuron (Rosenblatt, 1957). The main function of the perceptron is to learn binary classifications.

The following description is based on Bishop (2006), Dreyfus (2005) and Kruse (2015).

A perceptron introduces a parametrized and bounded function, which can be conveniently represented graphically as shown in Figure 2.1. The output of the perceptron is defined through a combination of the inputs $x_i \in \mathbb{R}$, weighted by the parameters $w_i \in \mathbb{R}$, which are often called 'weights'. Therefore, the most frequently used potential v is a weighted sum of inputs with an additional constant term $w_0 \in \mathbb{R}$ called 'bias':

$$v = w_0 + \sum_{i=1}^{n} w_i \cdot x_i \tag{2.1}$$

Finally, the potential v passed to the activation function f (e.g. *binary step* – see Figure 2.2 for possible activation functions), which is typically non-linear and produces the output y of the perceptron.

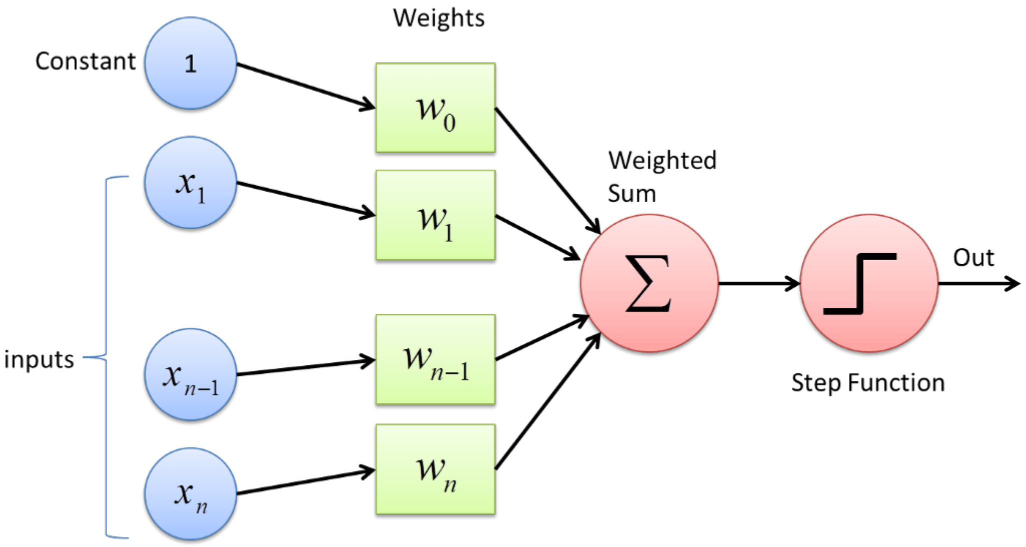


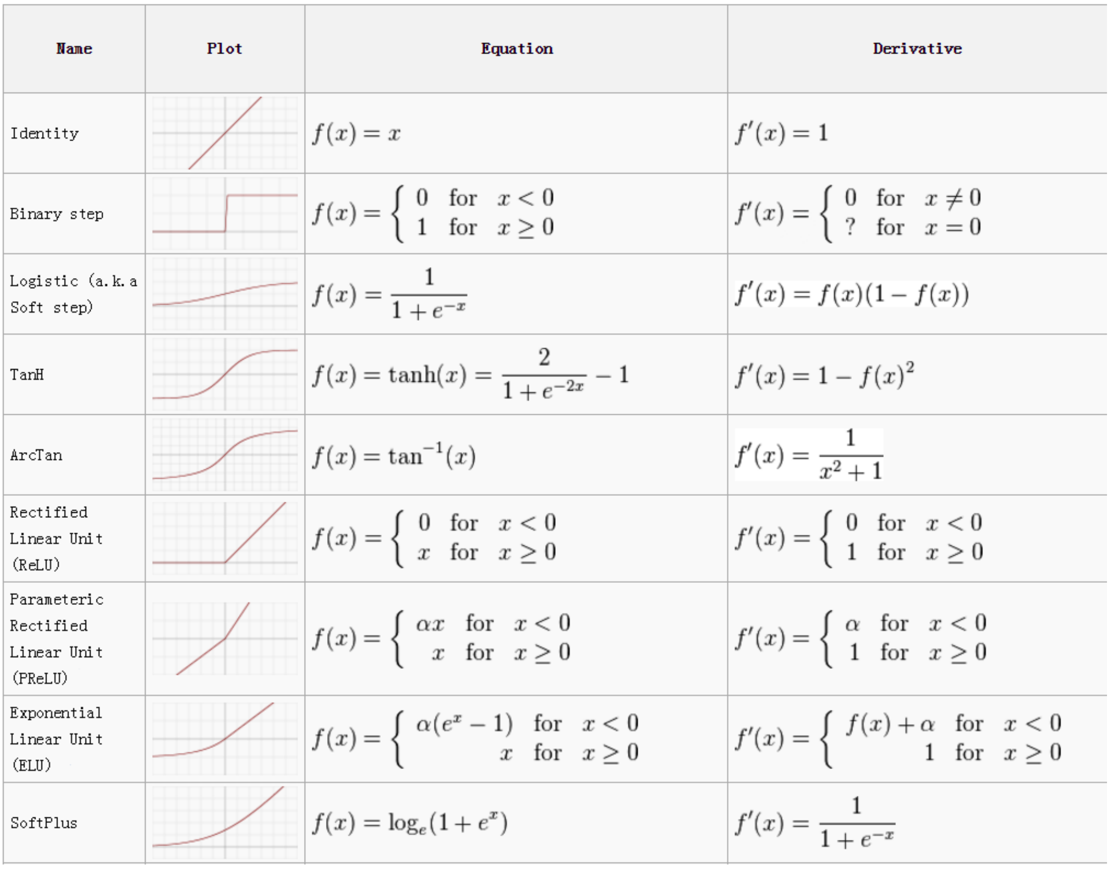
Figure 2.1.: Single perceptron. Graphic taken from Sharma (2017b).

Summing up together, a perceptron defines a function $y : \mathbb{R}^n \to \mathbb{R}$ of:

- an input vector $\tilde{x} \in \mathbb{R}^n$ and a constant given as a vector $x = \begin{pmatrix} 1 \\ \tilde{x} \end{pmatrix}$,
- a weight vector $\tilde{w} \in \mathbb{R}^n$ and a bias $w_0 \in \mathbb{R}$ given as a vector $w = \begin{pmatrix} w_0 \\ \tilde{w} \end{pmatrix}$,
- and an activation function $f : \mathbb{R} \to \mathbb{R}$, so that:

$$y(x) = f(w^T \cdot x) \tag{2.2}$$

Classification of x in one of two classes C_1 and C_2 happens due to the output of $y(x)$: If $y(x) > 0$, then $x \in C_1$; if $y(x) < 0$, then $x \in C_2$. Therefore, the equation $y(x) = 0$ defines the $(n-1)$-dimensional hyperplane separating the classes C_1 and C_2 in $\mathbb{R}^n$. By the classification algorithm, the weights represent the strength of a particular node or parameter, the bias value allows shifting of the activation function curve up or down.



Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x \neq 0 \\ ? & \text{for} \quad x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \dfrac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \dfrac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \dfrac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)		$f(x) = \begin{cases} \alpha x & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for} \quad x < 0 \\ 1 & \text{for} \quad x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \dfrac{1}{1 + e^{-x}}$

Figure 2.2.: Perceptron activation functions. Graphic taken from Sharma (2017a).

2.2. Multi-Layer Perceptron

Perceptron algorithm has significant limitations in classification, for example, the XOR-problem. Figure 2.3 illustrates the problem: As shown in the previous chapter, perceptron defines an $(n-1)$-dimensional hyperplane separating two classes, but it is impossible here to separate the classes using one line.

The solution for the problem is to connect layers of single perceptrons in a more powerful model – Multi-Layer Perceptron (MLP).

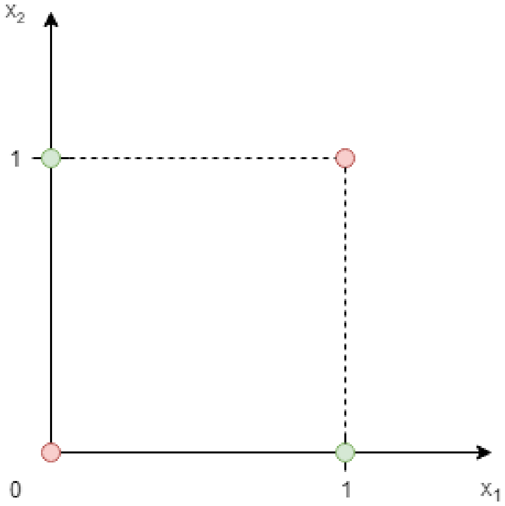


Figure 2.3.: The XOR-problem. The classes C_1 and C_2 (indicated as red and green circles respectively) are inseparable, when only one 1-dimensional line is used.

2.2.1. General structure

Here the output of one layer is used as the input for the next one (Figure 2.4).

The MLPs are usually defined through weight matrices $W^{(i)}$ for each layer U_i, which consist of connection weights between layers U_{i-1} and U_i. Thus, the computation of the input of any layer U_i can be simplified through:

$$\overrightarrow{net}_{U_i} = W^{(i)} \cdot \overrightarrow{in}_{U_i} = W^{(i)} \cdot \overrightarrow{out}_{U_{i-1}}, \tag{2.3}$$

and the function y of an MLP is defined recursively as:

$$\begin{aligned} y_1(x) &= f_1(W^{(1)} \cdot x) \\ y_i(x) &= f_i(W^{(i)} \cdot y_{i-1}(x)) \\ y(x) &= y_r(x), \end{aligned} \tag{2.4}$$

where

r	number of layers
y_i	output of the layer U_i
f_i	activation function of the layer U_i

As in a simple perceptron, the values of the output layer are interpreted as classification results. MLPs are able to classify non-linear data (such as XOR-problem), since they can define more than one hyperplane. MLP is only one type of the ANN (also known as *feedforward* network) alongside with *recurrent* (or *feedback*) networks. To sum up the description and purpose of the ANN, it computes a non-linear function of its inputs.

2.2.2. Training of an Artificial Neural Network

To fulfil the assigned task (e.g. classification) the ANN should be *trained*. In closer look training is the algorithmic procedure, which estimates the parameters (weights) of the neurons in a neural network. Training falls into three categories: supervised, unsupervised and reinforcement learning.

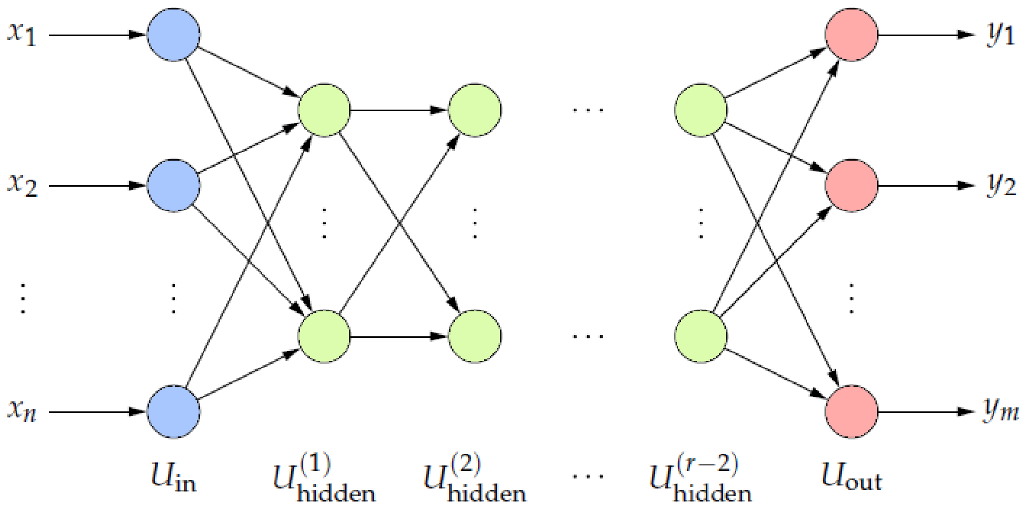

x_1
x_2
x_n
y_1
y_2
y_m
U_{in}
$U^{(1)}_{\text{hidden}}$
$U^{(2)}_{\text{hidden}}$
$\cdots$
$U^{(r-2)}_{\text{hidden}}$
U_{out}

Figure 2.4.: General structure of an r-layered Perceptron. Input layer is colored with blue, hidden layers with green and output with red. Graphic based on Kruse (2015).

Supervised learning

Such a task could be to 'learn' a specific non-linear function, which is known analytically or which is unknown, but a finite number of numerical values of the function are known. In such cases, *data*, which is a set of paired inputs and desired outputs (*labels*), is given by 'teacher' (therefore 'supervised') and the task is to produce the desired output for each input. The most common example of supervised learning is classification.

Unsupervised learning

The purpose of unsupervised learning is to find previously unknown patterns in a dataset, as a task for data visualization or analysis. Here the data is given without any labels, thus the network should 'discover' unknown patterns on its own, since no 'teacher' is present. In general, tasks for unsupervised learning are estimation problems such as clustering or self-organization.

Reinforcement learning

Reinforcement learning is usually defined through interaction with the environment, which reacts to actions taken by actors. The aim of the ANN is to learn the best-possible sequence of actions, whereby the long-term (expected cumulative) cost would be minimized (or reward maximized). Commonly the sequential decision-making tasks such as games or control tasks involve reinforcement learning.

2.3. Convolutional Neural Network

One of the goals of this work was to train a TTS system that will result in fast audio synthesizing. In order to fulfil the requirement and have a desirable model soon after start of this work it was decided to use a fully-convolutional architecture, which enables parallel computation and therefore trains faster than analogous models, employing recurrent networks (such as Shen et al. (2018)).

Convolutional Neural Networks (CNNs) have already shown impressive results in the image processing field, and now they are proceeding to be widely used in Natural Language Processing. Similar architecture was originally introduced under the name 'Time-Delay Neural Network (TDNN)' for phoneme recognition purposes by Waibel et al. (1987) and finally presented by Waibel et al. (1989).

Figure 2.5 shows the general structure of a CNN for image processing purpose, firstly presented in LeCun et al. (1989) and finally perfected for digits recognition by LeCun et al. (1998).

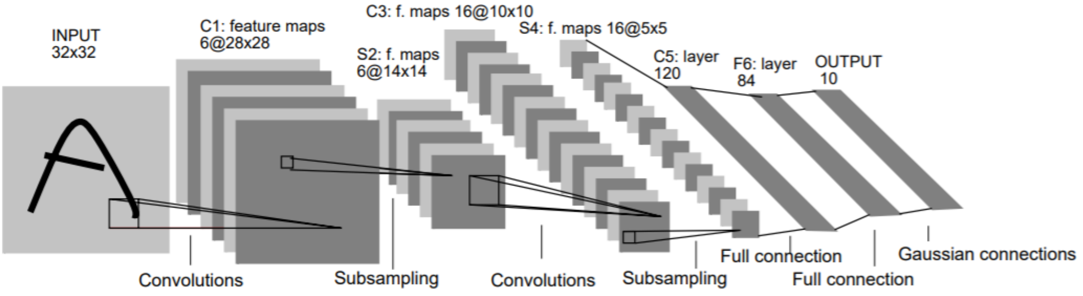
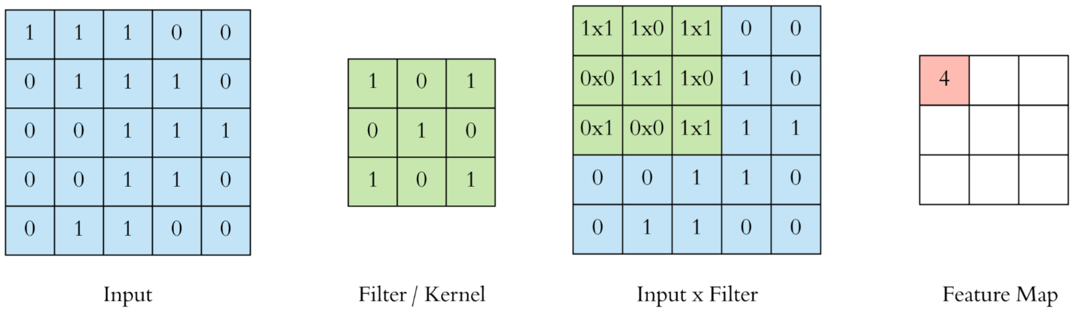


Figure 2.5.: Architecture of LeNet-5, a CNN for digits recognition. Graphic shows the general structure of a typical CNN, taken from LeCun et al. (1998).

Typically a CNN consists of multiple successive pairs of convolutional (*Cx*) and sub-sampling (*Sx*) layers followed by a regular neural network. Each layer produces several artefacts called *feature maps* of a certain size ($k@m \times m$, where k is a number of feature maps and $m \times m$ – its size), which are fed to the next layer afterwards.

Convolutional layer is the main building block of a CNN. In the process of convolution, a *filter/kernel* of a specific size slides over the input with a certain *stride* (length of a step – usually 1), producing feature map due to the element-wise matrix multiplication and summing of the result. The exemplary convolutional layer is illustrated in Figure 2.6. Usually, multiple convolutions on input are performed in each layer, each using a different filter and resulting in a distinct feature map (Dertat, 2017). In order to preserve the non-linearity of an ANN, the result of the convolutional layer could be passed through *ReLU*-function.



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

(a) Initial conditions for convolutional layer.

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

Input x Filter

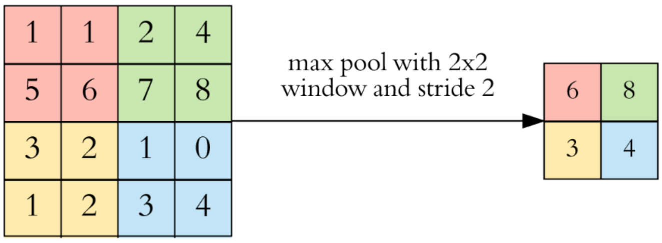
4		

Feature Map

(b) Process of convolution, producing a feature map.

Figure 2.6.: Example of a convolutional layer in a CNN. Graphic taken from Dertat (2017).

Sub-sampling or **Pooling layer** reduces the dimensionality of a problem, which lowers both training time and risk of overfitting (Dertat, 2017). During the pooling process, a pooling window of a specific size slides over the input with a certain stride, operating current values in the window. Typical examples are based on the operation: *max pooling* returns the maximum value, *average pooling* returns the average of all the values in the window. An example of max pooling is depicted in Figure 2.7. Pooling layers downsample each feature map independently.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Figure 2.7.: Example of a pooling layer in a CNN. Graphic taken from Dertat (2017).

Fully-connected layers follow the architecture of the above, allowing to train CNN as a common ANN described in Section 2.2. The output of the final pooling layer must be flattened before feeding to the fully-connected layers.

The main purpose of CNNs is to reduce input in a form that is easier to process and that still has original characteristics. Convolutional layers are used to extract local elementary features, which are then combined by the subsequent layers in order to detect higher-order distinctive features. Composition of several feature maps allows multiple features to be extracted at each location. Pooling layers decrease the computational power required to process the data, and a common ANN efficiently learns a non-linear combination of high-level features in order to correctly classify the input (LeCun et al., 1998).

CNNs could be applied to all types of data. In comparison to 2-D from the above (mainly visual data such as images and videos), it is possible to use CNNs for 1-D data (e.g. strings of characters or words). The key difference is the dimensionality of a kernel (one-dimensional) and how it slides across the data (only in one direction). 1-D convolutional filters are widely used in the Natural Language Processing field, especially in the TTS system described in this work.

2.4. Encoder-Decoder Network

NLP purposes (such as multi-language machine translation) require mapping of sequences to sequences, thus new end-to-end approaches to sequence learning are needed. Encoder-Decoder Networks, firstly introduced by Sutskever et al. (2014), provide such possibility. The ANN architecture converts the input sequence to a vector of a fixed dimensionality (using *encoder*), and then decodes the target sequence from this vector (using *decoder*). The original paper proposes the use of a special type of Recurrent Neural Network (RNN), namely, Long Short-Term Memory (LSTM), which are able to learn long-term dependencies. Nevertheless, it is possible to reduce training time and to outperform the accuracy of LSTMs through the application of architecture based entirely on CNNs (Gehring et al., 2017).

General function of a sequence-to-sequence model is to map a variable-length input with a variable-length output where the lengths of the input and output may differ.

Formally, the encoder processes an input sequence $X = (x_1, \dots, x_m)$ of m elements and returns state representations $Z = (z_1, \dots, z_m)$, called *encoder vector*. The decoder takes z_m and generates the output sequence $Y = (y_1, \dots, y_n)$ left to right, one element at a time (in RNN-based architecture). To generate output y_{i+1}, the decoder computes a new hidden state h_{i+1} based on the previous state h_i, an embedding of the previous target language word y_i, as well as last encoder state z_m (Gehring et al., 2017).

Fully convolutional architecture relies on CNNs instead of RNNs in order to compute intermediate encoder states Z and decoder states H. In addition, since CNNs accept as input only a fixed number of elements, padding with zero vectors at each layer is applied. CNN blocks operate with 1-D convolutional kernels.

Training involves both encoder and decoder networks. Target output could be used, while the probability $P(y_n \mid y_{n-1}, \ldots, y_1, X)$ should be maximized.

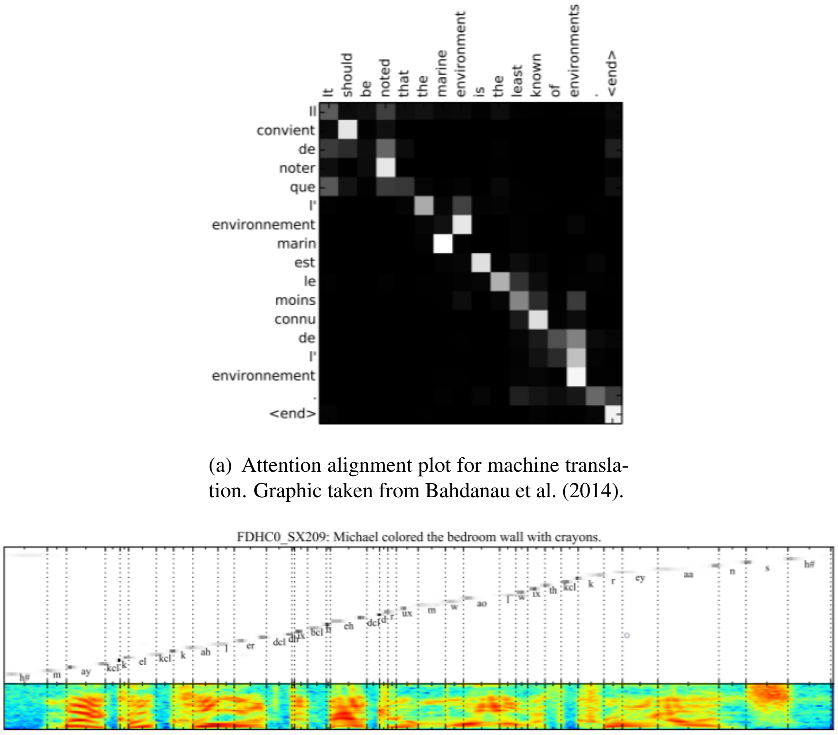
Attention Mechanism was introduced to encoder-decoder networks by Bahdanau et al. (2014) (for neural machine translation) and Chorowski et al. (2015) (for speech recognition), allowing them to achieve higher performance than traditional sequence to sequence models. Use of a fixed-length encoder vector is seen as a bottleneck in modern encoder-decoder networks, thus the attention mechanism provides a possibility to a decoder to automatically search parts of the source sequence that are relevant in predicting a target sequence in each step of the output generation (Bahdanau et al., 2014).

Formally, models without attention consider only the final encoder state z_m, with which the first encoder state is initialized (see Section 2.4). Architectures with attention keep the whole sequence of encoder hidden states Z, computing a *conditional input* (or *context vector*) c_i at each decoder time step (Gehring et al., 2017):

$$c_i = \sum_{j=1}^{m} \alpha_{ij} \cdot z_j, \tag{2.5}$$

where the α_{ij} (*weights* or *attention scores*) can be interpreted as a measure of how much the output at time i aligns with the input at time j. Attention scores are usually computed based on the *alignment model* a, which is parametrized as a simple feedforward neural network and trained with all other components of an encoder-decoder network, and normalized to be a distribution over input elements (Sutskever et al., 2014).

Attention provides an intuitive way to interpret and represent what the model is doing. This is done by visualizing the weights α_{ij} or more generally the alignment model a, as shown in Figure 2.8.



(a) Attention alignment plot for machine translation. Graphic taken from Bahdanau et al. (2014).

(b) Attention alignment plot for speech (phoneme) recognition. Graphic taken from Chorowski et al. (2015).

Figure 2.8.: Examples of attention visualization.

The x-axis and y-axis in Figure 2.8(a) correspond to the words in the source sentence (English) and the generated translation (French), respectively. Each pixel shows the weight α_{ij} of the annotation of the j-th source word for the i-th target word, in grayscale (from 0: black to 1: white). It can be seen that the alignment of words between English and French is largely monotonic, however, there is a number of non-trivial, non-monotonic alignments.

Figure 2.8(b) depicts the alignment of phonemes (transcription) to speech. The vertical bars indicate ground truth phone location, each row of the upper image indicates frames selected by the attention mechanism to emit a phone symbol. Attention in Automatic Speech Recognition (ASR), as well as in TTS sequence-to-sequence models, should produce mainly monotonic left-to-right alignments with strong weights along the diagonal of the matrix, which would reflect the good performance of the model.

Attention mechanism is also applicable to encoder-decoder networks based on CNNs (Vaswani et al., 2017).

2.5. Vocoder

Vocoder is an algorithm to estimate a signal (e.g. raw audio waveforms) from its representation by predicting vocoder parameters. Usually, mel-spectrograms or spectrograms derived from modified Short-Time Fourier Transform (STFT) are used as representations. Other parameters such as fundamental frequency, spectral envelope or aperiodic parameters could serve as well.

Algorithms could be as well hand-engineered (*Griffin-Lim* or *WORLD* vocoder) as in the form of a deep neural network (*WaveNet* vocoder).

Griffin-Lim algorithm converts spectrograms to time-domain audio waveforms by iteratively estimating the unknown phases (Griffin and Lim, 1984).

As **WORLD** vocoder parameters a boolean value (whether the current frame is voiced or unvoiced), a fundamental frequency value (if the frame is voiced), the spectral envelope, and the aperiodic parameters are predicted (Morise et al., 2016).

WaveNet vocoder treats linear- or mel-scale log-magnitude spectrograms as vocoder parameters. The WaveNet is trained using ground-truth mel-spectrograms and audio waveforms (van den Oord et al., 2016).

There are also numerous alternatives for high-quality vocoders in the literature (Ping et al., 2018).

3. Methods and algorithms of speech synthesis

Generally, all of the speech synthesis systems are divided into two parts: frontend and backend. Frontend considers itself with parameters and resources that are language-specific, and the methods and models used in backend are language independent. Mostly the backend influences the quality of synthesized speech and its characteristics. Therefore, TTS systems are classified based on the type of synthesis methods used in the backend.

The following chapter describes the most common technologies in the speech synthesis field.

3.1. Traditional TTS technologies

The two primary technologies in synthetic speech generation are concatenative synthesis and formant/articulatory synthesis.

3.1.1. Concatenative systems

Concatenative synthesis uses high-quality audio clips of different lengths, combining them to form a speech. These audio units must be pre-recorded from a single speaker before the synthesis occurs.

The most important aspect of concatenative synthesis is to find the correct unit length. Depending on the type of units used for concatenation, there are mainly three types of concatenative synthesis, namely, (1) domain-specific synthesis, (2) diphone synthesis, and (3) unit selection synthesis (Rao, 2019).

Domain-specific synthesis

The main unit in domain-specific synthesis systems is word or phrase. The limitation of the variety of utterances to be produced is defined through a particular domain, where the system will be used (transit schedule announcements, weather reports, etc.). This simple to implement technology is not generally-purpose and can only produce utterances, which are preprogrammed.

Diphone synthesis

Diphones are sound-to-sound transitions occurring in the language (opposite to phones that are any distinct speech sound, whilst diphones could be described as any adjacent pair of phones). For example, Spanish has about 800 diphones and German has about 2,500. The main advantage of the approach is the small size of the speech database, but diphone synthesis frequently suffers from audible distortions when two diphones that are not compatible with each other are concatenated (Rao, 2019).

Unit selection synthesis

Unit selection synthesis generates the speech by concatenating the natural speech segments selected from a large speech database. The database contains multiple instances of each unit, which can be of different length: half-phones, phones, diphones, triphones, demi-syllables, syllables, morphemes, words, phrases, or even sentences (Rao, 2019). The choice of the unit depends on the nature of language and the target application (Hunt and Black, 1996).

Unit selection synthesis can provide highly natural and intelligible speech if a large well-optimized corpus is available. By today many unit selection systems have achieved a high quality of the synthesized speech: *MITalk* (Allen et al., 1987), *CHATR* (Black and Taylor, 1994) and *Festival* (Taylor et al., 1998).

The drawbacks of the model are low flexibility of voice characteristics and high complexity of the unit selection algorithms itself because of the co-articulatory effects between adjacent units (in order to provide more natural speech).

Phonemes

One of the most commonly used units in the unit selection synthesis is phoneme. A phoneme is one of the units of sound that distinguish one word from another in a particular language. For example, the widely known set of phonetic transcription codes ARPAbet comprises 39 phonemes (not counting lexical stress) of General American English.

The CMU Pronouncing Dictionary contains over 134,000 words and their pronunciations based on the ARPAbet phonemes set and was extensively used during this work (CMU, 2014).

Concatenative speech synthesis introduces a simple and effective way to produce intelligible and natural-sounding speech (although some works consider the generated speech emotionless (Saxena, 2017)). In addition, this method is less complex than the others from the computational point of view (Datta, 2018), but the overhead needed to develop a robust system, which usually takes seven to eight months (Saxena, 2017), is the main weakness of this approach alongside with high memory capacity.

3.1.2. Source-filter systems

As of Fant (1960): "The speech wave is the response of the vocal tract filter system to one or more sound sources. This simple rule, expressed in the terminology of acoustic and electrical engineering, implies that the speech wave may be uniquely specified in terms of source and filter characteristics". This statement is a foundation and an actual interpretation of both formant synthesizers and articulatory synthesizers.

Formant synthesis

A formant synthesizer is a source-filter model in which the source models the glottal pulse train and the filter models the formant resonances of the vocal tract (Smith, accessed 08/01/2019). Based on the analysis of the speech data the human experts derive a set of rules, which specify directly the formant frequencies and bandwidths as well as the source parameters (Birkholz, 2017).

Hereby a formant stands for the spectral shaping that results from an acoustic resonance of the human vocal tract (Titze, 1994). The primary source of sound, the voicing produced by the vibration of the vocal cords is imitated by vibrating reed and the turbulence noise produced due to the pressure difference across a constriction (Datta, 2018).

Articulatory synthesis

Articulatory synthesis methods are also modelled after the human speech production mechanism, but these synthesizers, on the other hand, determine the characteristics of the vocal tract filter by means of a description of the vocal tract geometry (i.e. movement of articulators of speech production mechanism with time) and place the potential sound sources within this geometry (Birkholz, 2017). The main issue of this rule-based approach is the complexity in deriving articulatory rules for speech production.

The source-filter speech synthesizers produce mostly artificial, robotic-sounding speech, nevertheless the intelligibility of the speech can be very high. Due to the absence of the speech samples database, formant/articulatory synthesizers are usually smaller programs than concatenative systems, which makes them easy to use as embedded systems. The other advantage is that the source-filter systems have complete control of the output speech, hence a wide variety of changes could be made to the output voice (such as intonations, emotions, and tones). Nevertheless, the main disadvantage of the approach is the high complexity of rules that determine the speech synthesis.

3.2. HMM-based synthesis

HMM-based synthesis (or in other words Statistical Parametric Speech Synthesis (SPSS)) uses the principle similar to formant synthesis which involves parameterization of speech during the training phase and reconstructing the speech from the parameters during the synthesis phase (Rao, 2019). The parameters here are the frequency spectrum (vocal tract), fundamental frequency (voice source), and duration (prosody) of speech. In this system, the HMMs are used in an unified framework to model the parameters and also to generate speech waveforms based on the maximum likelihood criterion (Bishop, 2006).

The HMM-based generated speech is smooth and intelligible. The SPSS has high parametric flexibility and it can be adapted to different voice quality, speaking style, and emotion by using a small amount of target speech data. The amount of speech data required for training the SPSS is very small (Rao, 2019). Nevertheless, many artefacts are resulting in muffled speech or buzzing and noisy sound (Saxena, 2017).

3.3. Speech synthesis using Deep Neural Networks

The main problem of the HMM-based synthesis is that the certain features for speech synthesis are hard-coded by humans, but they are not necessarily the best features to synthesize the speech (Saxena, 2017). This is where deep learning could help: recently DNNs have also been extensively used for modelling the parameters in SPSS, as they can extract computer-readable language features better than humans.

The intelligibility and naturalness of the recent TTS systems that involve DNNs are achieving very high levels so that the produced speech is nearly human-like. Furthermore, many experiments based on TTS systems were made, which allow further development of speech synthesizers and include modern techniques such as voice conversion and reproduction of dialects or emotions (Tits et al., 2019) and (Zhang et al., 2019).

The next chapter will reveal state-of-the-art techniques in deep neural speech synthesis.

4. Text-to-Speech system

4.1. General structure

The following description is based on Ping et al. (2018).

The Deep Voice 3 architecture consists of three components (see Figure 4.1 for further details):

- **Encoder**: A fully-convolutional encoder, which converts textual features to an internal learned representation;
- **Decoder**: A fully-convolutional decoder, which decodes the learned representation with a convolutional attention mechanism into a low-dimensional audio representation (mel-scale spectrograms) in an autoregressive manner;
- **Converter**: A fully-convolutional post-processing network, which predicts final vocoder parameters (depending on the vocoder choice) from the decoder hidden states.

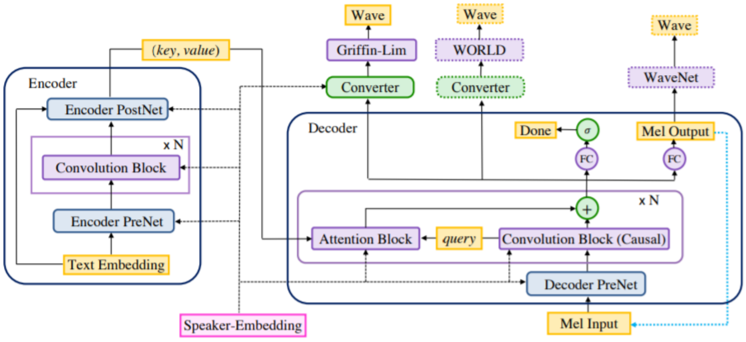


Figure 4.1.: Deep Voice 3 architecture. Graphic taken from Ping et al. (2018).

Text preprocessing is done in order to normalize input and eliminate mispronunciations, and therefore improve performance. During preprocessing all characters are upper- (or lower-) cased, all intermediate punctuation marks are removed and every sentence is ended with a punctuation mark. This model is capable of joint representation of characters (graphemes) and phonemes, thus the pronunciation could be modified to correct common mistakes.

Encoder network firstly converts textual inputs into trainable vector representations, h_e, through the embedding layer. After projecting to target dimensionality embeddings h_e processed through a series of convolution blocks to extract time-dependent text information. Attention key vectors h_k of the embedding dimension are created, and afterwards, attention value vectors are computed from attention key vectors and text embeddings, $h_v = \sqrt{0.5}(h_k + h_e)$. The key vectors h_k are used by each attention block to compute attention weights, whereas the final context vector is computed as a weighted average over the value vectors h_v.

Decoder network generates audio in an autoregressive manner by predicting a group of r future audio frames conditioned on the past audio frames. Mel-scale log-magnitude spectrogram is chosen as the compact low-dimensional audio frame representation.

The decoder network starts with multiple fully-connected layers with rectified linear unit (ReLU) non-linearities to preprocess input mel-spectrograms. The following convolution blocks generate the queries used to attend to the encoder's hidden states over the attention block. Lastly, a fully-connected layer output the next group of *r* audio frames and also a binary "final frame" prediction (indicating whether the last frame of the utterance has been synthesized).

Converter network takes as inputs the activations from the last hidden layer of the decoder, applies several convolution blocks, and then predicts parameters for downstream vocoders.

Several vocoders can be used with described architecture such as *Griffin-Lim* (Griffin and Lim, 1984), *WORLD* (Morise et al., 2016) or *WaveNet* (van den Oord et al., 2016) vocoders.

The overall objective function to be optimized is a linear combination of the losses from the decoder and the converter.

More detailed information could be found in Appendix A.

4.2. Datasets comparison and training

Much of the success of the TTS system depends on the choice of the dataset used to train the system. It must be neither too large (since greater size does not usually mean better performance) nor too small (to avoid overfitting and to train the model properly) and be of a good quality, which means that each utterance has no misspoken or removed or added words (in accordance with text) and no noise or buzz could be heard in each utterance. Each dataset should contain *data*: Text sentences in form of strings of characters with punctuation marks at the end ('.' or '?' or '!'), and corresponding *labels*: Waveform Audio File Format (WAVE) files containing utterance. Thus, most of the datasets found during this work should have been formatted and adapted to restrictions above. The full list of used datasets and their properties resides in Appendix B.

For each dataset an attention alignment plot from the encoder-decoder network will be given for a test phrase as in Section 2.4, demonstrating achieved performance of the model. Figures 4.2 to 4.8 are built on the same principle: the x-axis shows encoder hidden states, which are attended by the decoder in each decoder timestep (on y-axis). Attention weights are normalized between 0 and 1 (coloured legend bar on the right). Legend also displays a number of iterations performed on the training of the model before synthesizing of the particular utterance. Since speech corresponds to text subsequently (word by word), clear and smooth strictly-diagonal attention alignments should indicate better speech intelligibility, thus are expected. Nevertheless, the final decision is reached only after listening to the synthesized utterances.

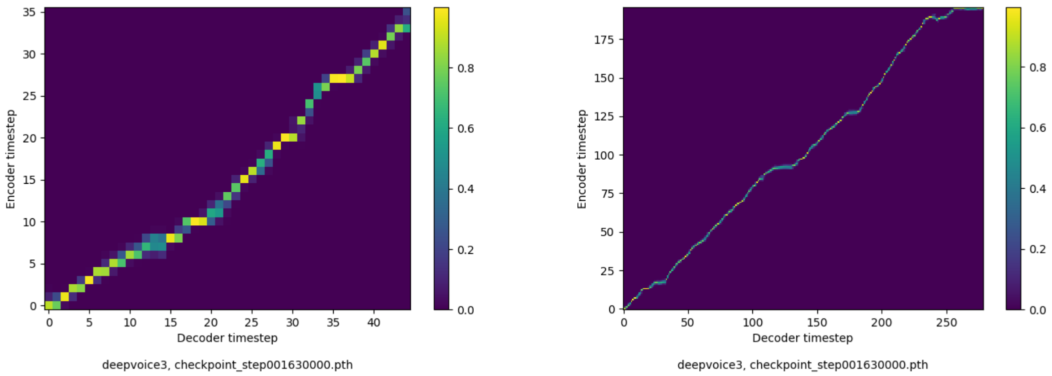
4.2.1. English model

LJSpeech

The most commonly used dataset for training TTS systems is the LJSpeech dataset. This dataset contains 24 hours of good labelled female speech. After two and a half days of training, a relatively good result was obtained. The synthesized speech was comprehensible, in addition the word flow was human-like. However, some of the phonemes in particular words resulted in robotic pronunciation. The other problem is that the WAVE data files in this dataset have a maximal length of 10 seconds, consequently, the synthesized utterances are good until the 10th second, and then there is only noise. Figure 4.2 illustrates performance of the model for several test sentences.

CMU_ARCTIC

The system established itself as functioning good, thus the male speech datasets were searched next. The first found was the CMU_ARCTIC database with seven relatively small datasets of about 1132 sentences and 51 minutes of speech in each set. These datasets are very good labelled and contain no errors in pronunciation. Nevertheless, the main problem is that they are too small, so after training the



(a) *"Armar, please give me the hammer"*. Mainly clear alignment results in high speech intelligibility.

(b) *"Once upon a time there was a dear little girl who was loved by everyone who looked at her, but most of all by her grandmother, and there was nothing that she would not have given to the child"*. Clear and smooth alignment except for several flat surfaces, indicating pauses or mispronunciations. After 10 seconds of synthesized audio only noise was produced, as seen at the end of the alignment curve.

Figure 4.2.: Attention alignment plots for the LJSpeech model.

synthesized speech was not understandable because of poor quality. Conclusion: The dataset is too small for training of the TTS systems based on ANN only using this set (see Figure 4.3(a)).

It is also possible to train the ANN as a multi-speaker model, therefore it was trained next using three male speech sets (all from the CMU_ARCTIC database). The result was the same as by training on the single-speaker dataset: Poor quality and almost incomprehensible speech. The reason for this result might be the following: all the datasets contain the same pronounced sentences so that the speech does not generalize during training (see Figure 4.3(b)).

The other possibility is to fine-tune a pre-trained model and adapt it to data from another dataset. First, the pre-trained model on the LJSpeech dataset was adapted to the raw data from one of the CMU_ARCTIC datasets (namely, to *'bdl'* dataset). In that way, the good male speech was obtained, which was much better than the speech produced by the model trained on the CMU_ARCTIC dataset from scratch. The voice was not smooth, but the comprehensibility of the synthesized speech was almost acceptable. Figure 4.3(c) visualizes the improvement of model performance. Since this approach demonstrated good results, all of the following experiments were conducted as an adaptation on the best model received – the one trained on the LJSpeech dataset.

The other problem is that all utterances from the CMU_ARCTIC database have durations shorter than 6 seconds, which results in poor quality of the synthesized speech after 6-8 seconds. To solve this problem the following approach was applied: Every four sentences (and corresponding audio files) from 'bdl' dataset were merged into one so that the mean duration of the utterances was approx. 10-12 seconds. Then the pre-trained on the LJSpeech dataset model was adapted to this merged 'bdl' dataset. The quality of the synthesized speech has risen a little as well as the quality of longer sentences (more than 8 seconds). The issue of this approach is that now the shorter sentences are synthesized as 8 seconds-audios, where the first 3 seconds refer to the actual sentence and the rest is noise. Figure 4.3(d) illustrates the problem for a short test phrase.

The next idea to solve this problem was to merge not all of the utterances, but make a distribution of short and long audio files: 25% was not merged at all (mean duration 3 seconds), 25% of the new audio files consist of two files merged into one (mean duration 6 seconds), 25% consist of three audio files (duration of 9-10 seconds) and the rest 25% are four audio files merged into one (12 seconds and above). Then the adaptation of the pre-trained model on this new dataset was performed. The resulting quality was the same as in the last experiment, but now the sentences of all lengths could be synthesized without any unnecessary information or noise (see Figure 4.3(e) for a short test phrase as well as Figure 4.3(f) for a longer one). This model has the best quality of the synthesized speech between all of the models trained before.

Nevertheless, other issues were remaining to be solved:

- The audio files were clipped too soon at the end (the ending of the last word was not understandable);
- The quality of the speech must be improved.

LibriSpeech

The problem of almost all the datasets above is that they are too small to successfully train the ANN, so it was decided to train the TTS system with the significantly greater dataset: LibriSpeech (approx. 100 hours). This is a multi-speaker dataset, containing data from 251 speakers. Data was taken only from 126 male speakers, resulting in approx. 50 hours of transcribed speech. Since the maximal duration from single speaker (among all speakers) does not exceed 25 minutes, the best results could be only achieved from the training of a multi-speaker model on this set. LibriSpeech is an ASR Corpus dataset, which are often much larger but tend to be less clean, as they typically involve background noises. ASR Corpora could be 'reversed' and used for TTS purposes. The training lasted for more than a week (approx. 7-8 days), but, contrary to assumptions, the results were unacceptable because of very poor quality, as the speech was not understandable at all (see Figure 4.4).

The main reason for this result is the purpose of the dataset: ASR datasets may contain errors in transcription, and due to the size of the dataset amount of the errors could be very high, resulting in poor quality of speech. The other issue, associated with this dataset, refers to the training routine: multi-speaker models tend to converge to a lesser degree than analogous single-speaker models because of the different voice and pronunciation manners of the speakers.

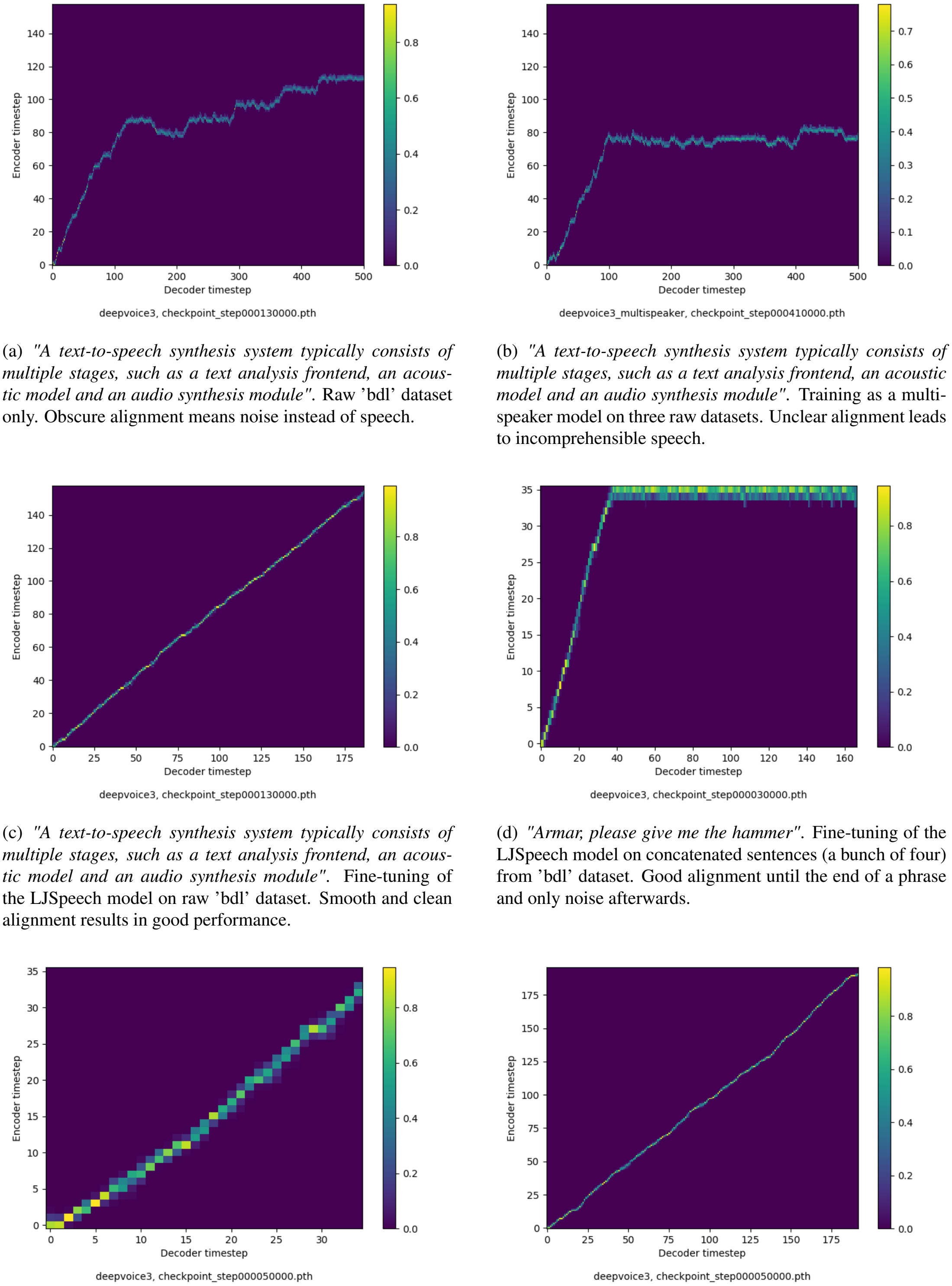
Mozilla Common Voice

Common Voice by Mozilla is an open-source multi-language speech database, based on a crowd-sourcing platform. As accessed on 04/02/2018, the database contained voice utterances from 26744 speakers, resulting in almost 804 hours of speech, however, it is still growing.

As Common Voice is a crowd-sourcing platform, the evaluation of the recorded speech (as well as recording itself) is done by users. Thus, all the data is divided into three parts: validated (significantly more up-votes than down-votes), invalidated (significantly more down-votes as up-votes) or other (not evaluated yet or the same number of up- and down-votes). For training of the TTS system only 'validated' and 'other' utterances from two best-sounding speakers with maximal duration (4.38 hours for male and 1.91 hours for female) were taken. Models were trained for approx. one and a half days and one day respectively.

Both models (male and female) were equally good comprehensible, but still, some of the words could be mispronounced in both models. As for naturalness, both models sounded robotic or even unpleasant (the male voice). The other issue was that the male speech was produced in diverse volume: Louder short utterances and more silent long utterances. Figure 4.5 illustrates the main problems in the attention learning of both models.

Since this database is crowd-based, it can have errors in pronunciation or completely wrong utterances even after the evaluation and disposal of the invalidated data. That is the case with male speaker data above: Although some of the erroneous utterances were excluded (even those that were incorrectly marked as 'validated'), it is nearly impossible to clean the whole dataset. The consistency of data was also disrupted, that is why more stable female dataset produced better results.



(a) *"A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module"*. Raw 'bdl' dataset only. Obscure alignment means noise instead of speech.

(b) *"A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module"*. Training as a multi-speaker model on three raw datasets. Unclear alignment leads to incomprehensible speech.

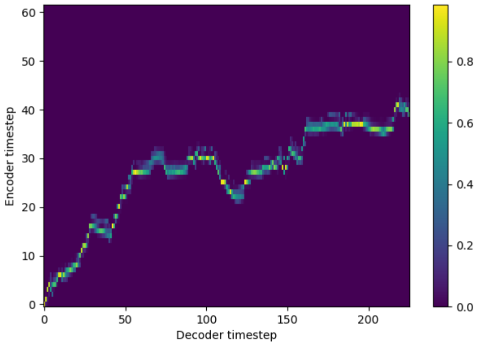
(c) *"A text-to-speech synthesis system typically consists of multiple stages, such as a text analysis frontend, an acoustic model and an audio synthesis module"*. Fine-tuning of the LJSpeech model on raw 'bdl' dataset. Smooth and clean alignment results in good performance.

(d) *"Armar, please give me the hammer"*. Fine-tuning of the LJSpeech model on concatenated sentences (a bunch of four) from 'bdl' dataset. Good alignment until the end of a phrase and only noise afterwards.

(e) *"Armar, please give me the hammer"*. Fine-tuning of the LJSpeech model on concatenated sentences (mix bundle) from 'bdl' dataset. Smooth alignment without noise.

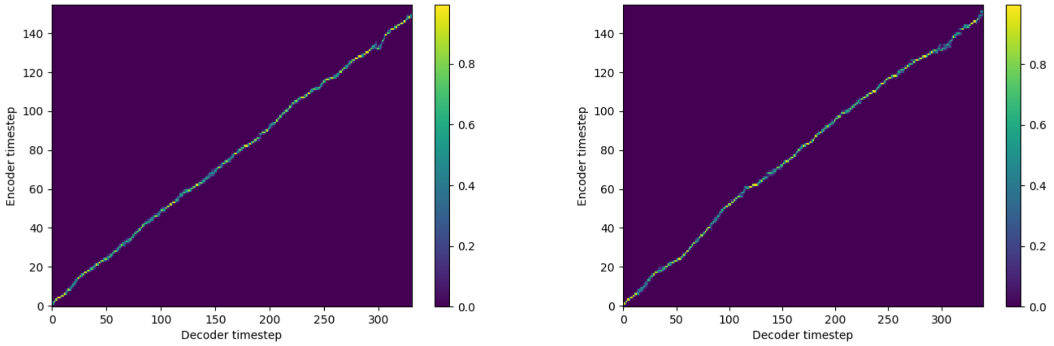
(f) *"Once upon a time there was a dear little girl who was loved by everyone who looked at her, but most of all by her grandmother, and there was nothing that she would not have given to the child"*. Same as (e). Smooth and clean alignment without noise.

Figure 4.3.: Attention alignment plots for the CMU_ARCTIC model.



deepvoice3_multispeaker, checkpoint_step001770000.pth

Figure 4.4.: Attention alignment plot for the LibriSpeech model. The test sentence is: *"Generative adversarial network or variational auto-encoder"*. Unclear alignment implies that attention is not learned properly, thus the output contains only noise.



deepvoice3, checkpoint_step000610000.pth

deepvoice3, checkpoint_step000250000.pth

(a) Male speech model. Mainly smooth alignment except for several instabilities, which match mispronunciations.

(b) Female speech model. Clean and smooth alignment except for several noisy areas and disruption in the end, which indicates unclear mumbling.

Figure 4.5.: Attention alignment plots for the Mozilla Common Voice database. The test sentence in alignments is: *"Printing, in the only sense with which we are at present concerned, differs from most if not from all the arts and crafts represented in the Exhibition"*.

VoxForge

Another open-source crowd-sourcing platform for collecting transcribed speech is VoxForge. The highest quality and the maximal duration among all speakers were achieved by *'ralfherzog'*, resulting in approx. 9 hours of transcribed speech.

The first training attempt (two days of training) did not derive any comprehensible speech, in fact, only unclear noises were produced (see Figure 4.6(a) for attention alignment). Since the dataset consists of multiple parts, which were recorded at different times each, it was decided to investigate reasons, causing bad results in synthesizing. In order to do that, the generalization of the model was verified: Some of the utterances were taken from each part of the dataset and subsequently trained together in small batches of 16 utterances. As the TTS system could 'learn by heart' that small amount of utterances, the 'unlearned' ones, which could not be reconstructed and properly synthesized after training, were later excluded from the dataset.

After excluding the inconsistent parts of the dataset the overall duration decreased to 7.35 hours. The following training increased the quality of the synthesized speech considerably, which made it the basis for the final model of this work.

The only problem remaining was clipping at the end of the synthesized utterances. Trimming of leading and trailing silence from an audio signal (by the *librosa* package (McFee et al., 2019)) is based on a threshold (in Decibel) below reference, through that silence is detected and cut. This technique allows a significant reduction of training time and produces more natural speech, however, more quiet phonemes at the end of the sentence could be recognized as silence and trimmed. Figure 4.6(b) illustrates the problem. In order to avoid that another tool (*SoX* (Navarrete, 2009)) was used, herewith silence at the beginning was trimmed completely and all periods of silence longer than 0.5 seconds were trimmed down to only 0.5 seconds long, which solved the clipping problem, as shown in Figure 4.6(c).

The main issue of the VoxForge dataset is the monotonous speaking style of the speaker: It results in more understandable speech, meanwhile naturalness of the voice remains low, which makes it not human-alike.

The final version of the English TTS model, trained for approx. a day on this dataset will be evaluated in section 5.2.1.

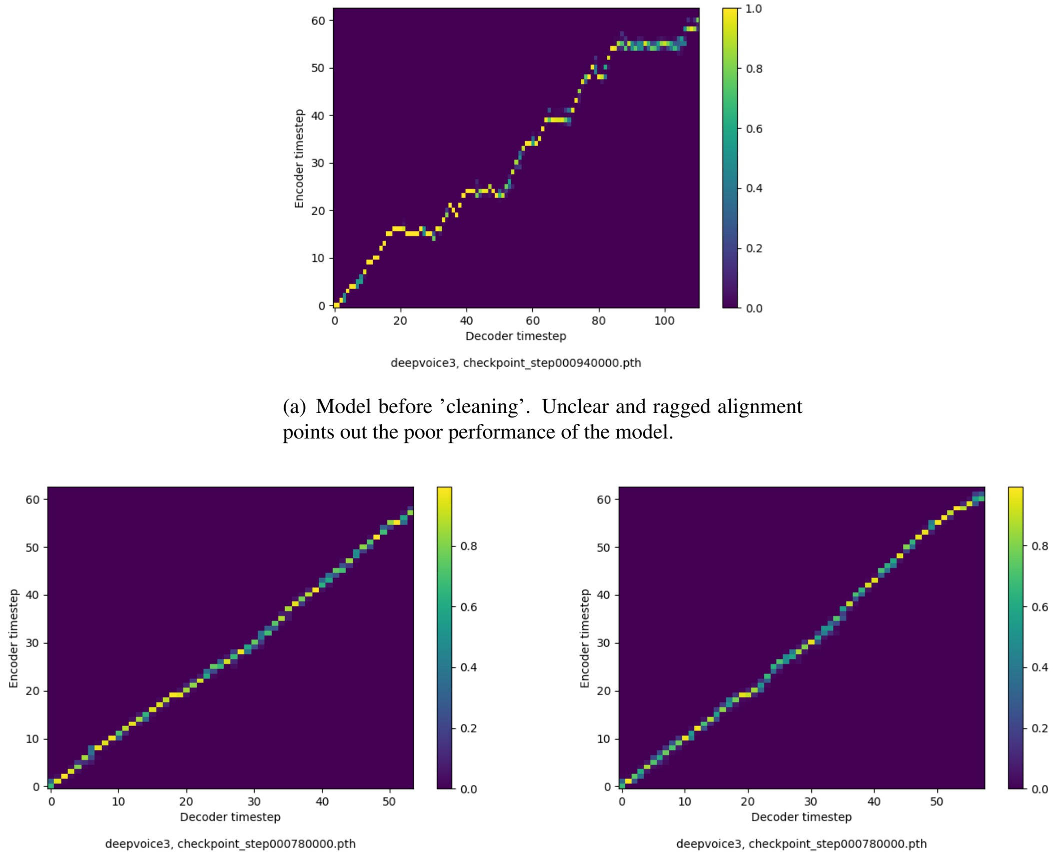
4.2.2. German model

Finding a non-English speech dataset, even of an inappropriate quality, is a common issue for almost every language. Still, some existing German datasets were found and the system was trained on them during this work.

Since the *Deep Voice 3* TTS system implementation was not capable of training of German model, a new frontend should be written. First of all, Germanic umlauts (such as 'Ä', 'Ö', 'Ü', 'ä', 'ö' and 'ü') and Eszett ('ß') were added to the set of graphemes. Conversion of a string of text to a sequence of Identifiers (IDs) corresponding to the symbols in the text and vice versa (sequence of IDs back to a string) was also added.

All required operations of an input text (such as punctuation and lower-case) were written for German frontend. In order to extend the usability of the system normalization of German numbers such as years, currencies, ordinal numbers, etc. (using *num2words* tool (Dupras, 2019)) was added. All of the above should act as a cleaner of German text and thus increase the quality of the trained on that text model. It is suggested to perform the expansion of abbreviations in the frontend since this version of the TTS system does not operate with acronyms.

After the first training attempts, it was determined that the system overfits the training data and thus could not produce test speech correctly. One of the ideas was to complicate training data using pronunciation dictionary for German language (as in English model). *CMU Sphinx* includes almost 32,000 German words and their pronunciations, therefore it was decided to use this pronunciation dictionary (CMU, 2019). During preprocessing words with 50% probability were substituted with their pronunciations, which significantly increased the quality of the synthesized speech. 66 representations of phonemes were added to the set of valid characters.



(a) Model before ’cleaning’. Unclear and ragged alignment points out the poor performance of the model.

(b) Model after ’cleaning’ and use of the *librosa* tool. Performance has risen considerably, although synthesized utterance is clipped in the end, as decoder did not attend all encoder states during the activity of attention mechanism.

(c) Model after ’cleaning’ and use of the *SoX* tool. Smooth and clean strictly-diagonal alignment indicates better performance and no clipping in the end. Nevertheless, noisiness could be seen (at some points), which can negatively affect naturalness of the synthesized speech.

Figure 4.6.: Attention alignment plots for the VoxForge ’ralfherzog’ dataset (English model). The test sentence in all alignments is: *"For the twentieth time that evening the two men shook hands"*.

LibriVox

One of the ideas for the dataset to train the TTS system was the use of audiobooks: They usually derive skilled and understandable pronunciation of the speech, and the amount of data available is huge. The main problem in this approach is the need for complex preprocessing, namely, in a splitting of the whole audiobook in small chunks (e.g. sentences).

The first German dataset found was a transcription of three free public domain audiobooks, read by a female volunteer. The books were already split in small chunks of data (mainly sentences or small parts of the sentence), therefore no data preprocessing was needed. The amount of data totals to 16 hours of transcribed speech of good quality.

Six days of training of the TTS system delivered poor results in intelligibility as well as in naturalness. Only the beginning of each sentence was pronounced right (but still in a robotic-sounding manner), all the other words were mispronounced or even omitted, resulting in sound interferences. Encoder-decoder network attention alignment for this dataset could be seen in Figure 4.7.

The outcome could depend on the dataset itself: Audiobooks should not correspond to 100% to the text, as only the main sense and some of the special words of the sentence must be delivered to the listener, thus the mispronunciations in spoken speech could occur. In addition, the audiobook itself could be split in an unfortunate manner during preprocessing, which could interfere with the final result.

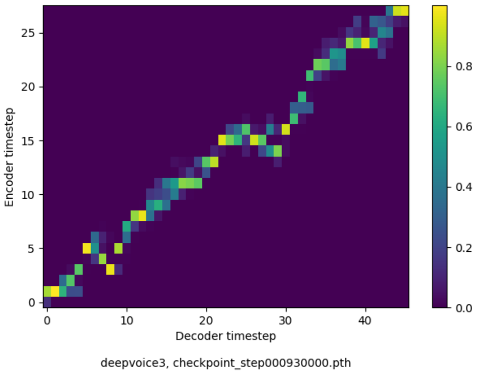


Figure 4.7.: Attention alignment plot for the LibriVox model. The test sentence is: *"Das ist gar nicht möglich"*. The alignment curve is ragged and noisy, although it tends to be strictly-diagonal and smooth in certain small areas, which allows the model to produce separate understandable words.

Mozilla Common Voice

Common Voice speech database contains speech data for 29 languages (state of 04/07/2019), including German. The German database contains voice utterances from 2195 speakers, resulting in approx. 140 hours of transcribed and evaluated speech.

After short revision of the utterances from speakers with the longest duration (one hour and more) the following conclusion was made: Some of the recordings lack in quality of sounding or pronunciation, some were recorded by several (male or female) speakers, which would negatively influence training of the TTS system as well as the results. A group of people recorded utterances from one account so that the database contained data from several speakers indicated as data from one.

To keep preprocessing times smaller a decision was made not to include this database in training routine. Datasets from certain speakers remain promising, but they need to be investigated closely to detect and extract all inconsistencies.

VoxForge

As Common Voice, VoxForge has datasets for several languages, with German among them. From all datasets in this database four were taken for further investigation, and finally, data from the same speaker as for the English model (*'ralfherzog'*) with duration of almost 24 hours was used for training.

The first results, obtained after almost three days of training, showed that the ANN was too complex for the data and written frontend, thus the overfitting happened, as data from training set was precise reproduced on opposite to text data from test set, which derived only noise (see Figures 4.8(a) and 4.8(b)). In order to fix it, a pronunciation dictionary for German language was introduced to the frontend, which was elaborately discussed in section 4.2.2. In addition, some inconsistent utterances were excluded from dataset, decreasing the overall duration to 23.3 hours.

The second attempt illustrated considerable progress in both intelligibility and naturalness of synthesized speech. To remove clipping of the synthesized audios the same tool and approach as for the English VoxForge dataset was used (*SoX*), which reduced overall duration to only 13.5 hours. Figures 4.8(c) and 4.8(d) visualize the improvement of the model.

Final three days-long training provided best results for German TTS model: The synthesized speech is highly intelligible and relative natural, nevertheless the absence of any intonation and long pauses between words in several utterances make the sounding of this model unnatural for some users. The intelligibility is also interfered by accidental repetition of random phonemes, as illustrated in Figures 4.8(e) and 4.8(f).

This version of the German TTS model is considered as final and will be evaluated in section 5.2.2.

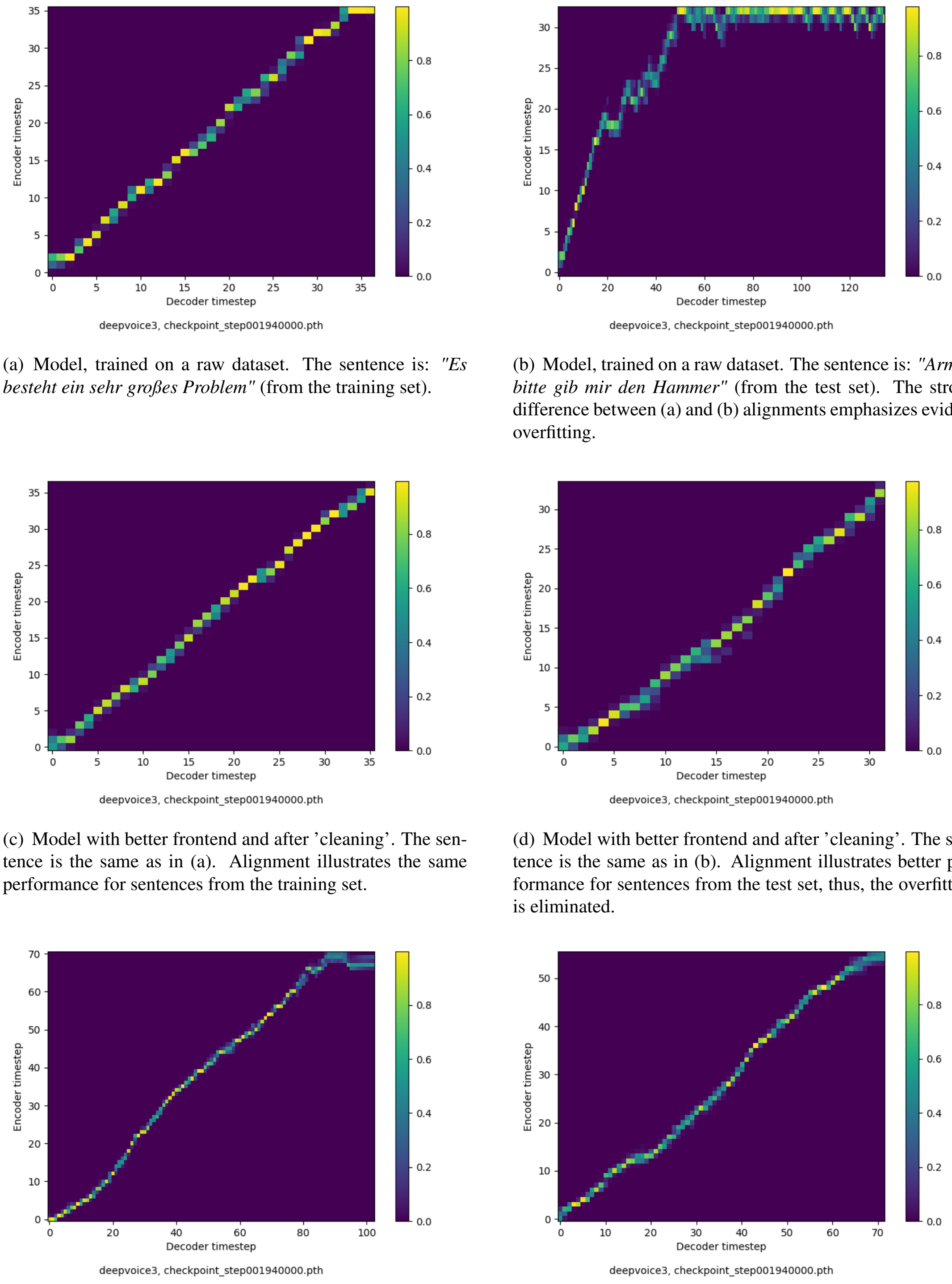
4.3. Synthesis

Speech was synthesized using the Nvidia Tesla K80 GPU card. The test sentences for validating of synthesis performance were:

1) "March was windy and dark" (5 words) for English model and "März war windig und dunkel" (5 words) for German model;
2) "Portals are used in many computer games to instantly travel from one place to another" (15 words) for English model and "Portale werden in vielen Computerspielen verwendet, um sofort von einem Ort zum anderen zu reisen" (16 words).

Latency (time to initialize the model) lies between 2.9 and 3.1 seconds, duration of synthesizing of the first sentence is about 1.5 seconds; of the second sentence: about 2.0-2.1 seconds (same for both models).

It is also possible to synthesize speech using only the CPU, however, the performance of such operation was not measured during this work.



(a) Model, trained on a raw dataset. The sentence is: *"Es besteht ein sehr großes Problem"* (from the training set).

(b) Model, trained on a raw dataset. The sentence is: *"Armar, bitte gib mir den Hammer"* (from the test set). The strong difference between (a) and (b) alignments emphasizes evident overfitting.

(c) Model with better frontend and after 'cleaning'. The sentence is the same as in (a). Alignment illustrates the same performance for sentences from the training set.

(d) Model with better frontend and after 'cleaning'. The sentence is the same as in (b). Alignment illustrates better performance for sentences from the test set, thus, the overfitting is eliminated.

(e) Final version of the model. The test sentence is: *"Im Jahr 1998 lebten dort 2000 Bürger"*. Alignment shows that the model is capable of dealing with written numbers.

(f) Final version of the model. The test sentence is: *"Straßen? Wo wir hingehen, brauchen wir keine Straßen"*. Alignment is smooth and clean, which allows the best intelligibility of synthesized speech. Nevertheless, several parts of the alignment curve are noisy, which could lead to unobtrusive mispronunciations or repetitions.

Figure 4.8.: Attention alignment plots for the VoxForge 'ralfherzog' dataset (German model).

5. Evaluation and results

There are several different evaluation techniques for testing synthetic speech and the TTS system itself. All of them can be divided into two major parts: Objective/acoustic measures and user evaluation (Cryer and Home, 2010).

The objective sense of acoustic testing reveals itself in measuring whether or not an utterance from a synthetic voice acoustically matches the same utterance in human speech. One example of acoustic measures are the statistical methods such as Root Mean Squared Error (RMSE), which estimates the difference between sound contours on a time axis (Cryer and Home, 2010).

Although objective testing is more efficient than subjective, due to the complexity of measurement and in order to match up the TTS system with listener perceptions the user evaluation was chosen in this work.

5.1. User evaluation

User evaluation is often done by users who are ultimately going to use it. Measures, delivered by user testing, can be divided into two main categories: performance measures (such as intelligibility) and opinion measures (such as naturalness and comprehensibility).

5.1.1. User studies criteria for TTS systems

Many works aimed at synthetic speech testing (Grimshaw et al. (2018) or Murthy et al. (2014)) distinguish between three main criteria for speech evaluation: intelligibility, comprehensibility, and naturalness.

Intelligibility refers to the accuracy with which each word is pronounced so that a normal listener can understand the spoken word or phrase (Murthy et al., 2014). The simplest method to analyze intelligibility of synthetic speech is to let respondents listen to an utterance and then write down all that they have heard. According to transcription accuracy percentage, the Word Error Rate (WER) (equation (5.1)) as a metric is used:

$$WER = \frac{S+D+I}{N}, \tag{5.1}$$

where

S number of substitutions
D number of deletions
I number of insertions
N number of words in a sentence

The overall score will be calculated as the average percentage of WER among all participants.

Intelligibility analysis requires special test data: Semantically Unpredictable Sentences (SUS). SUS are syntactically normal, but semantically abnormal sentences. As first proposed in Benoît (1990) and further discussed in Benoît et al. (1996), SUS allow evaluation of intelligibility at word level, without any influence made by the redundancy of the language. Benoît et al. (1996) discuss the process of constructing a test set of sentences, recommending to use a wide variety of sentence structures and mini-syllabic words to further reduce contextual cues (Cryer and Home, 2010).

Comprehensibility means up to what extent the message received is understood (Murthy et al., 2014). Hereby the understandability of the context should be measured, and not the recognition level of each word (in contrast to intelligibility). The testing method used here is the Mean Opinion Score (MOS), which gives a numerical indication of speech quality. Evaluators have to rate the comprehensibility of an utterance played on a five-point Likert scale (from 1 – very difficult to understand to 5 – very easy to understand). MOS then is the arithmetic mean of the mean scores given by each evaluator (equations (5.2) and (5.3)):

$$MOS_j = \frac{\sum_{i=1}^{N} s_{ij}}{N} \qquad (5.2) \qquad\qquad MOS = \frac{\sum_{j=1}^{M} MOS_j}{M}, \qquad (5.3)$$

where

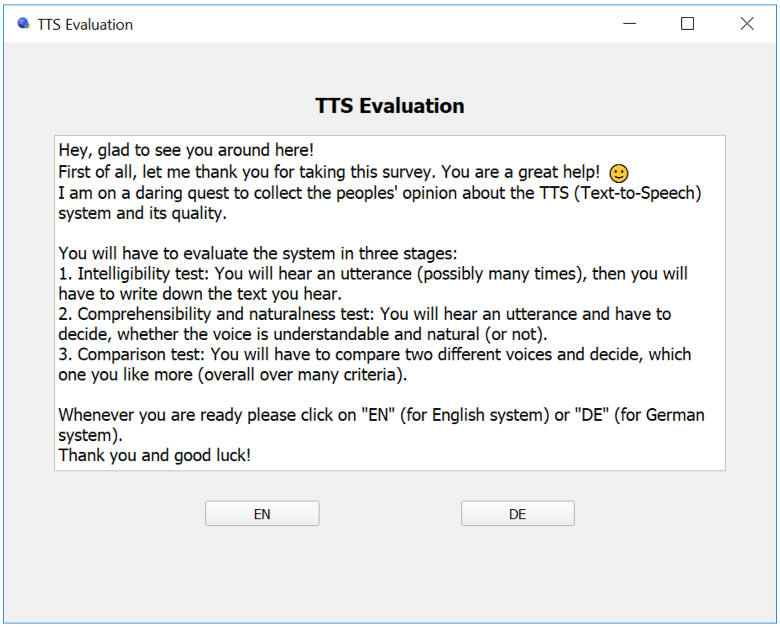
s_{ij} score of j-th evaluator for i-th sentence (from 1 to 5)
N number of sentences
M number of evaluators

Test data for comprehensibility evaluation should cover different variations and different lengths and should originate from different areas.

Naturalness describes whether the synthetic sound is indistinguishable from human speech and how close it is to the human voice (Murthy et al., 2014). The same test data and method (MOS – see equations (5.2) and (5.3)) as in the comprehensibility testing are used here, herewith respondents should rate sentences in terms of naturalness of the voice (from 1 – very unnatural to 5 – very natural).

5.1.2. Conducting of user testing

TTS system evaluation of this work is divided into three stages for the English model and into two stages for the German model. All user testing methods are gathered together in one Graphical User Interface (GUI) (as seen in Figure 5.1), created using the PyQt framework (Riverbank Computing, 2019).



TTS Evaluation

Hey, glad to see you around here!
First of all, let me thank you for taking this survey. You are a great help! 🙂
I am on a daring quest to collect the peoples' opinion about the TTS (Text-to-Speech) system and its quality.

You will have to evaluate the system in three stages:
1. Intelligibility test: You will hear an utterance (possibly many times), then you will have to write down the text you hear.
2. Comprehensibility and naturalness test: You will hear an utterance and have to decide, whether the voice is understandable and natural (or not).
3. Comparison test: You will have to compare two different voices and decide, which one you like more (overall over many criteria).

Whenever you are ready please click on "EN" (for English system) or "DE" (for German system).
Thank you and good luck!

Figure 5.1.: Start of evaluation session.

First stage

The intelligibility of both models (English and German) was evaluated in the first stage through fifteen (15) sentences, whereas for each heard utterance the corresponding transcript should be written and submitted (see Figure 5.2).

]

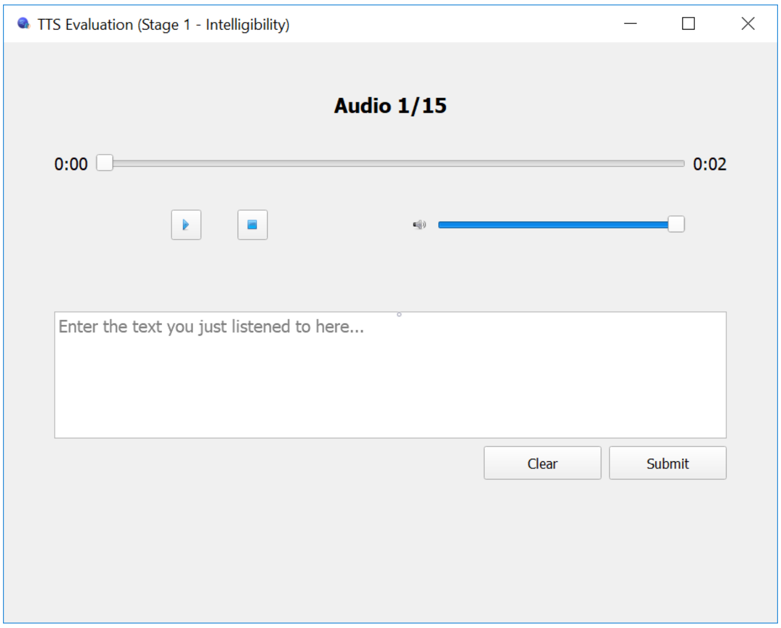

TTS Evaluation (Stage 1 - Intelligibility)
Audio 1/15
0:00
0:02
Enter the text you just listened to here...
Clear
Submit

Figure 5.2.: First stage of evaluation session – Intelligibility.

The sentences used here are ten (10) SUS and five (5) Harvard sentences (for English model)/Marburg (for German model) sentences.

The SUS are based on structure suggested by Gibbon et al. (1997). There are five groups of sentences with two sentences in each group:

1. Subject – Verb – Adverbial;
2. Subject – Verb – Direct object;
3. Adverbial – Transitive verb – Direct object (imperative sentence);
4. Q-word – Transitive verb – Subject – Direct object;
5. Subject – Verb – Complex direct object.

The Harvard sentences are a collection of sample phrases used for standardized testing (IEEE, 1969). The Marburg sentences originate from the Marburg Sentence Intelligibility Test (Brinkmann, 1974), which has been used for hearing tests with speech for many years in Germany.

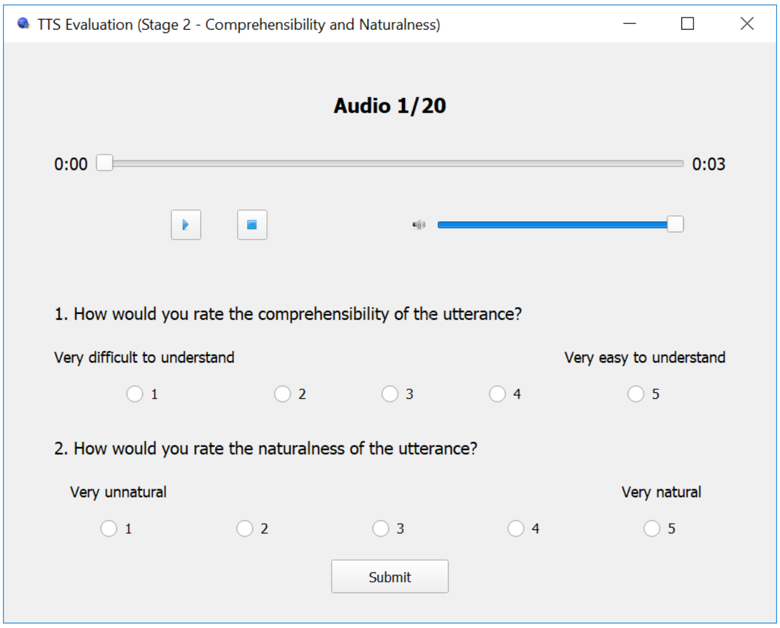
The test set contains both SUS and 'normal' sentences in order to reduce learning effects and study intelligibility of the system in both natural and extraordinary situations.

Second stage

In the second stage of evaluation, both opinion measures were gauged: Comprehensibility and Naturalness. For each of twenty (20) Harvard sentences (for English model)/Marburg or Wenker (for German model) sentences of different lengths, the two scores were gathered and submitted (see Figure 5.3).

The Wenker sentences were designed to present key lexical and grammatical items that could be used to differentiate the many German dialects. The 40 final sentences were firstly introduced in Wenker (1880).

Different sentence lengths result in different utterance durations: Both models were evaluated through nine (9) utterances with duration shorter than three seconds and eleven (11) utterances with duration greater than three seconds in order to make a more detailed statement about people's opinions.


TTS Evaluation (Stage 2 - Comprehensibility and Naturalness)
Audio 1/20
0:00
0:03
Submit

1. How would you rate the comprehensibility of the utterance?

Very difficult to understand — Very easy to understand

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

2. How would you rate the naturalness of the utterance?

Very unnatural — Very natural

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Figure 5.3.: Second stage of evaluation session – Comprehensibility and Naturalness.

Third stage (only for English model)

In the third stage, the tested model was compared to four other modern neural TTS systems, whereby the decision on which model is overall better for people to use was made.

The complete list of sentences used through the process of evaluation could be found in appendix C.

5.2. Results

The results of the evaluation for both models are discussed in this section.

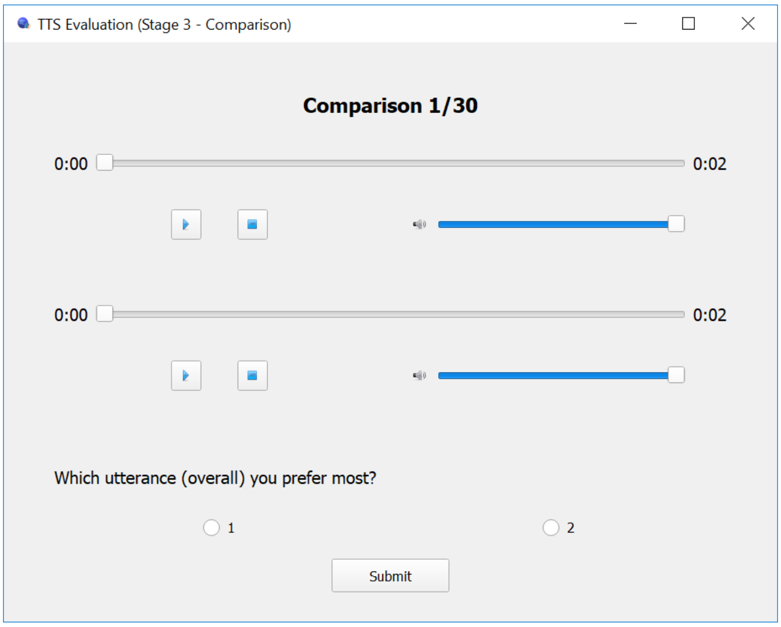
5.2.1. English model

The English TTS model was tested on seven respondents with diverse ethnicity and background knowledge in Text-to-Speech systems and Natural Language Processing in general. None of the respondents is a native English speaker, nevertheless, all of them can speak and understand English freely (B2 level on CEFR standard and above).

Table 5.1 shows the Word Error Rate (WER) for each of the respondents. It is noticeable that the WER for SUS is much higher than the corresponding WER for normal meaningful sentences. That means that it is more difficult for respondents to understand the sentence that makes no sense because of the absence of redundancy in such a sentence.

The English model produces almost 23.5% of WER overall, what can be seen as a high rate for a TTS system, nevertheless the WER for normal sentences reaches only 12% overall, which is an average rate for synthetic speech synthesizers. Besides that, some speakers (respondents 1 and 3) could achieve nearly zero WER, which is already a good result for intelligibility of synthetic speech.

Table 5.2 illustrates the MOS and its variations (MOS for utterances shorter or longer than three seconds) for comprehensibility of the English model. The overall score of 3.79 is more than average and it describes the pronunciation of the model as nearly easy to understand.


TTS Evaluation (Stage 3 - Comparison)

Comparison 1/30

0:00 0:02

0:00 0:02

Which utterance (overall) you prefer most?

○ 1 ○ 2

Submit

Figure 5.4.: Third stage of evaluation session – Comparison (only for English model).

Respondent	Mean WER (%)	WER for SUS (%)	WER for normal sentences (%)
1	12.45	18.67	0
2	39.63	46.94	25.00
3	18.62	26.69	2.50
4	19.06	21.92	13.33
5	23.97	30.41	11.11
6	18.27	21.39	12.02
7	32.37	38.00	21.11
Overall	23.48	29.15	12.15

Table 5.1.: WER for English model.

It is worth seeing again that the MOS depends on the length of the utterance: Utterances with length shorter than three seconds are more understandable than ones with greater lengths.

Respondent	All utterances	Utterances shorter than 3s	Utterances longer than 3s
1	3.00	3.11	2.91
2	2.25	2.67	1.91
3	4.40	4.56	4.27
4	4.30	4.22	4.36
5	3.40	3.22	3.55
6	4.45	4.67	4.27
7	4.70	4.89	4.55
Overall	3.79	3.90	3.69

Table 5.2.: MOS for comprehensibility of English model.

Table 5.3 demonstrates the MOS for naturalness and its variations (MOS for utterances shorter or longer than three seconds) of the English model. Naturalness of synthetic speech is seen worse than comprehensibility, achieving only 3.1 points overall.

Utterances shorter than three seconds are still being described more natural than utterances with longer length (3.33 vs. 2.91 points), which makes the system usable for short English sentences.

Respondent	All utterances	Utterances shorter than 3s	Utterances longer than 3s
1	1.05	1.11	1.00
2	1.00	1.00	1.00
3	3.80	4.33	3.36
4	4.50	4.56	4.46
5	3.30	3.56	3.09
6	4.00	4.33	3.73
7	4.05	4.44	3.73
Overall	3.10	3.33	2.91

Table 5.3.: MOS for naturalness of English model.

Table 5.4 shows number of achieved by each model votes in the third stage of evaluation. The model trained during this work features good performance against other open-source TTS models (such as DC-TTS (Tachibana et al., 2017) and Deep Voice 3 (Yamamoto et al., 2018)), which means that the model was preferred over others in more than a half of comparison steps. The Tacotron 2 (Shen et al., 2018) and Sample Efficient Adaptive Text-to-Speech (Chen et al., 2018) demonstrated best results in this stage of evaluation, however, both models use internal speech datasets, which must maintain higher quality standards than the other open-source speech datasets.

The evaluation of the English model indicates that all of the criteria (Intelligibility, Comprehensibility, and Naturalness) could be further improved in future work. Usually, TTS systems are analysed by native speakers, which was impossible in this work. Evaluation done by native speakers could significantly decrease mean WER and increase MOS for comprehensibility. Naturalness of the synthesized English speech should be further advanced. Nevertheless, the third stage of evaluation demonstrated a relative high competitiveness of the trained model with modern TTS systems.

(a) Comparison to Tacotron 2 (max. 8 votes).

Respondent	Current model	Tacotron 2
1	0	8
2	0	8
3	1	7
4	3	5
5	0	8
6	1	7
7	0	8

(b) Comparison to DC-TTS (max. 12 votes).

Respondent	Current model	DC-TTS
1	6	6
2	0	12
3	6	6
4	8	4
5	3	9
6	8	4
7	3	9

(c) Comparison to Sample Efficient Adaptive Text-to-Speech (max. 4 votes).

Respondent	Current model	Sample Efficient Adaptive Text-to-Speech
1	0	4
2	0	4
3	0	4
4	1	3
5	0	4
6	0	4
7	0	4

(d) Comparison to Deep Voice 3 (max. 6 votes).

Respondent	Current model	Deep Voice 3
1	6	0
2	0	6
3	6	0
4	6	0
5	5	1
6	6	0
7	6	0

Table 5.4.: Number of votes, achieved by each of the English models during the comparison routine.

5.2.2. German model

The German TTS model was tested on eight respondents with diverse ethnicity and background knowledge in Text-to-Speech systems and Natural Language Processing in general. Some of the respondents are native German speakers (respondents 2 and 3), the others can speak and understand German freely (C1 level on Common European Framework of Reference for Languages (CEFR) standard and above).

Table 5.5 shows the WER for each respondent with a statement of whether the respondent is a native speaker or not. As seen on the table, the overall WER is much lower for the German model than the WER for the English model (14.21% vs. 23.48%). This follows from the fact that the German dataset was larger as well as there were native speakers as respondents by evaluation. Thus the WER is much lower by native speakers (less than 10% or could even reach nearly-zero values) as by other respondents.

Table 5.5 also illustrates the fact that the WER for meaningful sentences is more than twice lower than the same rate for SUS (7.29% vs. 17.67%), which could be also interpreted through redundancy. The WER for normal sentences of the German model could be easily compared to the other modern TTS systems.

Respondent	Native speaker	Mean WER (%)	WER for SUS (%)	WER for normal sentences (%)
1	no	14.60	17.40	9.00
2	yes	3.72	3.92	3.33
3	yes	8.13	8.20	8.00
4	no	20.35	23.02	15.00
5	no	16.57	22.36	5.00
6	no	22.13	29.20	8.00
7	no	15.80	21.20	5.00
8	no	12.37	16.05	5.00
Overall		14.21	17.67	7.29

Table 5.5.: WER for German model.

Table 5.6 further proves the high comprehensibility of the German model. The overall score of 4.00 is good, that means it was stated by many of the respondents that synthesized German sentences are easy to understand.

Also here the utterances with length shorter than three seconds are simpler and perceivable than the others with greater length.

Respondent	All utterances	Utterances shorter than 3s	Utterances longer than 3s
1	4.30	4.44	4.18
2	4.25	4.78	3.82
3	3.60	4.11	3.18
4	4.50	4.33	4.64
5	4.10	4.00	4.18
6	2.90	3.00	2.82
7	3.80	3.89	3.73
8	4.55	4.56	4.55
Overall	4.00	4.14	3.89

Table 5.6.: MOS for comprehensibility of German model.

In Table 5.7 the scores for naturalness could be seen. Unfortunately, the naturalness of the model with a mean overall score by 3.00 was rated worse than the English model, but still, the score is in the middle of scale.

Naturalness of the speech is decreasing with the rising duration of the utterance and length of the sentence.

Respondent	All utterances	Utterances shorter than 3s	Utterances longer than 3s
1	3.65	4.00	3.36
2	3.10	3.44	2.82
3	1.35	1.67	1.09
4	3.20	3.78	2.73
5	3.20	3.67	2.82
6	2.65	2.78	2.55
7	2.70	2.89	2.55
8	3.95	4.11	3.82
Overall	2.98	3.29	2.72

Table 5.7.: MOS for naturalness of German model.

The training datasets for the German model were larger, therefore the error rate by pronunciation is lower as well as pronunciation itself is more comprehensible. Thus the naturalness of the voice should be further improved. Evaluation done only by native speakers would lower the error rate, which asserts the possibility of use of the German model for its purposes.

After evaluation of both models the following conclusions could be made:

- The synthetic speech of the system is quite intelligible and comprehensible for both models, although the German model copes better with its goals and is overall better understandable by listeners;
- Robotic-style voice of both models was rated as unnatural by human listeners, therefore naturalness of the TTS system should be further improved;
- Intelligibility depends on the meaning of the sentence: absurd sentences are less intelligible than normal everyday sentences (because of redundancy);
- Comprehensibility and naturalness of the speech are much higher for small short sentences rather than for long ones (three seconds borderline);
- Native speakers are more suitable for listening tests and provide better results.

6. Conclusion and future work

6.1. Conclusion

During this work, various sequence-to-sequence TTS systems (for English as well as for German languages), based on ANNs, were trained in order to synthesize comprehensible and natural male speech. Thus, different state-of-the-art architectures of modern TTS systems were reviewed and afterwards, an encoder-decoder architecture, built on CNNs, was chosen for this specific task. This work also reviews alternative traditional and recent TTS techniques, which are not associated with neural networks.

For the purpose of this work multiple open-source natural human speech datasets were found and explored on applicability for training of the TTS system. Between all models, trained on the found speech datasets, two were chosen (one of English and one of German language) for the next evaluation stage. Both datasets belong to the same database: VoxForge, and are recorded by the same speaker ('ralfherzog'). The English model involves training on almost four hours of transcribed speech, which leads to convergence of the model after one day of training. The German model is trained on 13.5 hours speech dataset and converges after three days of training. Since modern DNNs-based TTS systems converge mainly after a week of training, the achieved by this work result is more than acceptable.

The final model evaluation was done by users. Hereby three user studies criteria were measured: intelligibility, comprehensibility and naturalness. Overall WER for the English model reaches 23.5%, meanwhile, respondents have rated comprehensibility for 3.8 (out of 5.0) points and naturalness for 3.1 points on average. The same values are 14.2% of WER for the German model, and 4.0 points were given for comprehensibility and almost 3.0 points for naturalness of the model. Evaluation proves that the greater overall duration of the German speech dataset positively affects the performance of the model since WER and MOS for comprehensibility are higher at the average as those for the English model. In their turn, both models achieve WER of approx. 10% for normal sentences, therefore the synthesized male speech is suitable for any TTS-required purpose and is competitive with high-quality speech produced by other modern TTS systems. Discovered male speech datasets could be also applied as a standard for Text-to-Speech goals alongside with well-known LJSpeech dataset. The only weak point of the trained models is naturalness of the speech, which should be further improved.

Speech synthesis proceeds relatively fast, although it could be accelerated a bit. Model is loaded quicker than in three seconds, a medium length phrase (of about 10 words) is produced in less than two seconds afterwards. Synthesizing on low resource computers (without the use of GPU) was not tested, nevertheless, it is possible to produce speech on such systems. However, performance may decrease a little. In that way, the majority of this work's goals were reached.

6.2. Future work

A number of suggestions could be derived from this work, related to the trained model itself, discovered datasets and development of a new speech dataset. All suggestions and assumptions will be made below.

Model

Several improvements of the TTS system for better model training could be undertaken in future work.

Firstly, the weak point that still remains in both models is the potential in producing speech utterances of different lengths. The model is unable to produce clean and understandable utterances with duration more than 10 seconds, which was closely discussed in Sections 4.2.1 and 4.2.1. This work suggests an approach to solve this issue, discussed and successfully tested in Section 4.2.1, namely, the concatenation of short audio files to generate new datasets with longer utterances. Thus, phrases of every length could be effectively synthesized. However, this technique was not used in the final version of the model.

English model is challenged by repeating words in phrases (e.g. "eleven eleven o'clock" cannot be produced correctly), thus more training data of this type could fix the problem.

Modern spoken German language contains a variety of Anglicisms, which usually are not included in a training dataset, thus the German model mispronounced every English word in a sentence (or pronounced it in a German manner). A possible solution could be to involve an English pronunciation dictionary in a preprocessing routine for the German model since the substitution of English words to their pronunciations corrects the model. It is also possible to establish a whole new pronunciation dictionary for most common Anglicisms spoken in a German manner, what would fix the problem, however, it is too complex.

Naturalness of synthesized speech of both models could be considerably increased by training on a perfectly aligned speech dataset. In addition, fine-tuning of pre-trained model and subsequent training on another dataset showed promising results in speech performance.

Speech synthesis performance can be also increased by producing WAVE files on-the-go: While future audio frames are still being predicted, previous frames can be already played.

Discovered datasets

Most of the success of a TTS system relies on the dataset chosen for training. Multiple datasets were found and validated through training and subsequent evaluation of the models during this work, nevertheless, several further datasets could be inspected for TTS purposes.

Mozilla Common Voice database is further developing. As accessed on 08/30/2019, the database has significantly grown, which resulted in over 1000 hours of speech for English and over 300 hours of speech for the German language. Appendix B mentions several promising datasets for TTS system training, moreover, the amount of data in each set could have risen considerably. In addition, data from new speakers could have been added.

This work provides several bits of advice on preparing of the Mozilla Common Voice datasets for training:

- Firstly, data should be organized by speakers in order to train an optimal single-speaker model;
- It is suggested to train only on data marked as 'validated', as 'invalidated' and 'other' sets could contain mispronunciations or even erroneous utterances;
- Even though the 'validated' set is verified by users, it is recommended to inspect the whole dataset for inconsistencies such as acoustic differences between utterances, mispronunciations or noisiness, and multiple speakers.

VoxForge 'ralfherzog' dataset provided the best results for both English and German models in this work, however, it could be further improved. During this work, several acoustic inconsistencies were detected (such as the different volume of recorded speech), which could negatively affect the quality of synthesized speech. It is suggested to inspect the dataset on further drawbacks. In addition, recorded speech in 'ralfherzog' dataset is monotonous, which could be seen as unnatural by some users.

German VoxForge database contains additional promising major speech datasets (such as 'guenter' and 'manu'), on which the TTS system can be trained and thus deliver better results.

Other datasets can be found since this work does not cover all existing open-source speech datasets.

It is recommended to use *SoX* instead of *librosa* tool for removing silences before each training. Thus, training time is reduced considerably and clipping at the end of the synthesized utterances is eliminated.

Development of a new dataset

Since every dataset listed above has its negative sides, this work suggests to develop new speech datasets for both English and German languages with following characteristics:

- All data should proceed exclusively from one speaker (female or male);
- The size of the dataset should not exceed 24 hours and not be less than 10 hours (after silence elimination). Since training time of three-four days is considered as short by this work, 15 hours is believed as an optimal overall duration for the dataset and can be freely raised to 20 hours;
- Additionally, the dataset should consist of all possible utterance durations in order to produce speech from text of any length. This work suggests to record utterances with a maximal duration of 20 seconds;
- All utterances must be perfectly aligned. Every misalignment or mispronunciation leads to an insufficient performance;
- All utterances should involve good intonation, common for a normal speech. Lack of intonation is seen as unnaturalness of voice;
- Training set phrases should be phonetically balanced and contain all language variations (such as numbers, abbreviations, sentence structures, names and borrowed from other languages words);
- Clear pronunciation and no acoustic interference (such as noise) should be maintained.

Such a dataset could significantly increase model performance and become a new standard in the TTS field.

Appendices

A. Deep Voice 3 architecture

Detailed architecture of the Deep Voice 3 model is shown in Figure A.1.

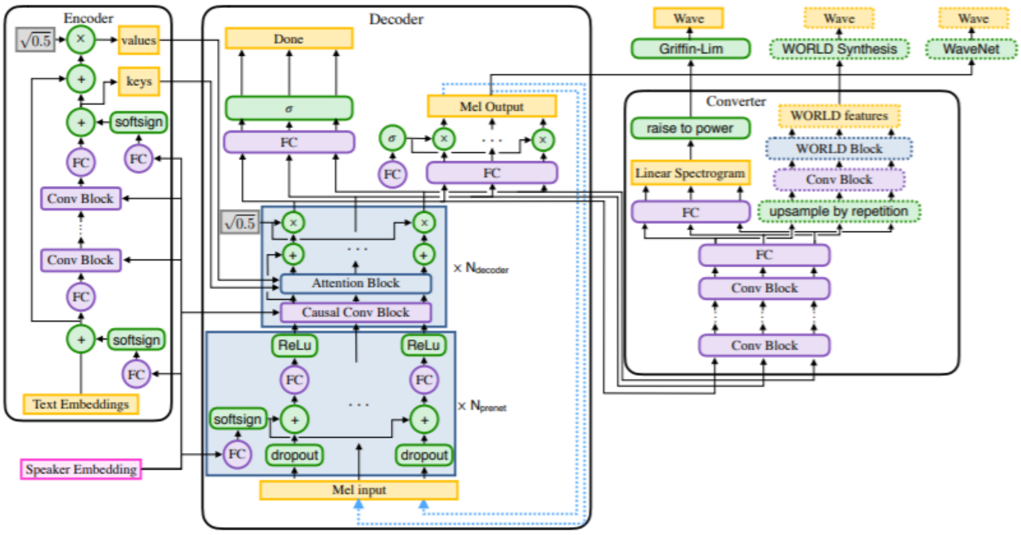


Figure A.1.: Detailed architecture of the Deep Voice 3 model. Graphic taken from Ping et al. (2018).

Every Convolutional Block for sequential processing in Figure A.1 consists of a 1-D convolution with a gated linear unit and a residual connection. Stacked convolutional layers can develop long-term context information in sequences without the use of RNNs, thus, allowing parallelism of computation.

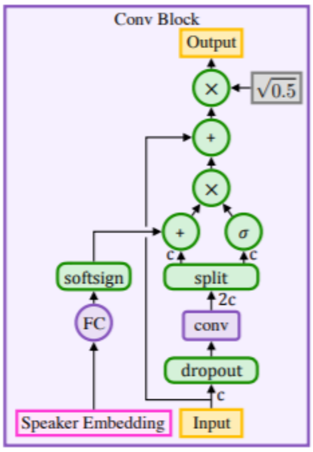


Figure A.2.: Convolutional block for sequential processing. Here c denotes the dimensionality of the input. Graphic taken from Ping et al. (2018).

B. Speech datasets

Datasets, used for training of the TTS system, and their properties are found in Tables B.1 (for English model) and B.6 (for German model).

Training was conducted with 16 GB Random-Access Memory (RAM) and on one of the following GPU cards:

1. Nvidia GeForce GTX Titan X *or* Nvidia GeForce GTX 1080 Ti (nearly similar performance);
2. Nvidia Tesla K80.

All the following tables mention explicitly the utilized GPU card for each dataset (and thus for each training routine).

Silence was removed before each training with tools such as *librosa* (always unless otherwise indicated) or *SoX* (explicitly indicated).

Quality refers to usability for the stated purpose, namely, training of the TTS system:

- 0 means impracticality of considering and using of the dataset (mostly because of multiple speakers or erroneous labels);
- 1 means partly possibility of using of the dataset, although it must be 'cleaned' (all inconsistencies must be removed) before training;
- $1-2$ are best datasets between all of quality 1, however, their applicability must be proven through training;
- 2 most useful datasets, since no inconsistencies were found.

Nevertheless, it is recommended to examine all datasets before training.

Databases with multiple datasets (such as CMU_ARCTIC, Mozilla Common Voice and VoxForge) are gathered together in separate tables. Tables for Mozilla Common Voice database contain only datasets with an overall duration of more than one hour.

Dataset	Sex	Duration (hours)	Duration after removing silence (hours)	Sample rate (Hz)	Training time (hours)	GPU card	Reference	Further information
LJSpeech	one female speaker	23.92	23.03	22050	61.5	1.	Ito (2017)	-
CMU_ARCTIC	three male speakers	-	-	16000	-	-	Kominek and Black (2003)	see Table B.2
LibriSpeech	126 male speakers	50.20	47.25	16000	≈180	2.	Panayotov et al. (2015)	multi-speaker model
Mozilla Common Voice	26744 male and female speakers	803.87	-	48000	-	-	Mozilla (2017)	see Tables B.4 and B.5
VoxForge	one male speaker	-	-	16000	-	-	VoxForge (2006)	see Table B.3

Table B.1.: English speech datasets.

Dataset	Duration (hours)	Duration after removing silence (hours)	Training time (hours)	GPU card	Note
bdl	0.85	0.78	8.3	1.	single-speaker
rms	1.10	0.99	10.5	2.	single-speaker
jmk	0.90	0.80	8.6	1.	single-speaker
bdl and rms and jmk	2.86	2.57	28.0	1.	multi-speaker

Table B.2.: CMU_ARCTIC speech datasets.

Dataset	Duration (hours)	Duration after removing silence (hours)	Training time (hours)	GPU card
ralfherzog (before ’cleaning’)	9.08	4.73	53.7	2.
ralfherzog (after ’cleaning’ + *SoX*)	7.35	3.92	24.7	1.

Table B.3.: VoxForge speech datasets.

Client ID	Intern speaker ID
e1bc135f88b68a2ef2662b6e8bc075a4417c72533096961a3cbf726d41e9b425c492a72827b2f86524b7881d56fde9dcea3130bf198ccb384f91bf88aec2efa6	26729
ad27250c8d473dfb74f58c467751ec1dea29752dc702631a769e66dcc48b1f15af75a514c0c01683c463a631a30dfccc0b7ef1cc71ecd71d48ba0a7d84dc55cf	26730
a12b2fd7f4d2fe8475c09e132784c8c25b1e7ab6706224d367c1010dfb49a06d6b7a25b900cdf110b0a1d1ed98112cd86366614f4a8c417bbe6264c2f013c2a2	26732
a964fbcb272707fc71749144dda3853f8a7e256d8c8e383c67839821d46322c7ebf30a397f918373361fc3ee6c387306caa7a76f569162a89e04ddd6ea0ddb5a	26733
3e275452aa87211d2b123764bfc72e077eb82ecde9789231f6173361801373e9d65c4eafa3fc994bd8ee7619e7249170e9e0cdbf387fc2687bb47b720af6ede9	26734
dfb21f3f7e5fe34d47dba5cea576a5aa5a5c999bd596a7aab5739558911bc3ba3d3dfe37014796c52ccfb2488bb9c32bfbd9561fc03274d1d5322e8b58f99905	26735
c4dbcbbfffb0f7b73e7ad4e9feca3878edf82f43ec17ef618c7a847bcc30faca432f802c344f64fb0afbc35ca70d3cf916bc1b496783dd9b21357d2678c8a5ba	26736
8722688699dc06b2714907cb6f0534f352edc8de08cfd3f066caef355a0f93df6b9f862e6adf92ffa440f519d457210d88aba82fbdfd846aa6746bc3cc4b27c8	26737
e4f4ff52da427a039c25d9a609663c1e92c436cebd8c331de170088c1d462c704297e524314ade67f40179d56f0dcd8b4f55abb0178c6a57d5a3bb14ee1e40a9	26738
960e0e9c5bc33cf05ac0e2d76a3cb93380869bcfe576809ffd7a725ff29018bb66c12c487be1113cb645072af9250eb5ecaa0e24e3ae9bbb41eef63633addeea	26739
b5cf5ab4217e0473de89c82de84e936fe9935b2a0df46631fb8be2eb504546b2958b689098f9293d9aec9b1757c2edf5e82c364c5237164e3de7f7f2cb15c29c	26740
2aa61a7ab7f29e2f25159af38779862289ea2512f80b3282d038dfb1abb9688f6ff1e364423b99d6cae24491d4322c9ffae1dbef51dbc95fd0c387d9fb7c0ae9	26741
e0a33c02ba5ff7aca57e9850a2d29d960ed6606c45befb7d3186880c9c08398d832c914cbf6ebe78ab204b8f2ebfedff8941d3c92515879e8cb06bb07c9db932	26742
335e3199eeb4394e44d5be9ca1cf0733842b392efc7636f6d8d737558167ed3a21493ecf45fb80c4344fc91218e1041387a91edb17f5a54c785541aa8b3a5cd0	26743

Table B.4.: Mozilla Common Voice speaker IDs alignment.

Dataset	Sex	Duration (hours)	Duration after removing silence (hours)	Training time (hours)	GPU card	Quality	Note
26729	female	1.11	-	-	-	1	high amplitude of voice
26730	male	1.01	-	-	-	2	-
26732	female	1.37	-	-	-	1	too quiet
26733	male	1.66	-	-	-	2	-
26734	female	1.15	-	-	-	2	-
26735	male	1.29	-	-	-	2	-
26736	male	1.57	-	-	-	2	-
26737	male	3.06	-	-	-	0	acoustic problems; multiple speakers
26738	female	1.54	-	-	-	2	second-best between female
26739	male	1.82	-	-	-	2	-
26740	female	1.91	1.22	24.6	2.	2	best between female
26741	male	3.10	-	-	-	1	acoustic problems
26742	male	4.38	2.69	41.6	2.	2	best between male
26743	female	3.18	-	-	-	1	acoustic problems

Table B.5.: Mozilla Common Voice speech datasets.

Dataset	Sex	Duration (hours)	Duration after removing silence (hours)	Sample rate (Hz)	Training time (hours)	GPU card	Reference	Further information
LibriVox	one female speaker	16.13	14.95	22050	145.3	2.	Park and Mulc (2018)	-
Mozilla Common Voice	2195 male and female speakers	139.84	-	48000	-	-	Mozilla (2017)	see Tables B.8 and B.9
VoxForge	four male speakers	-	-	16000	-	-	VoxForge (2006)	see Table B.7

Table B.6.: German speech datasets.

Dataset	Duration (hours)	Duration after removing silence (hours)	Training time (hours)	GPU card	Quality	Note
guenter	10.08	-	-	-	2	-
manu	9.45	-	-	-	2	-
openpento	2.03	-	-	-	0	erroneous labels
ralfherzog (before ’cleaning’)	23.52	12.67	66.5	1.	2	-
ralfherzog (after ’cleaning’ + *SoX*)	23.33	13.52	73.2	1.	2	-

Table B.7.: VoxForge speech datasets.

Client ID	Intern speaker ID
6d8a9d4d7068c73dd69ec4c9ee248e518a1fbeab7db90d21901a4b73bade4deecaae7bc1c96c0779d92af6bc2e59867cb2f0a7acca0b0f4051781668d1773cb0	2176
7e7c3c607551f259857163f8eef0bd98a71a9cbe25b5daa9e465eb3c2f9aa68f9342513ec719387677055804140 6fb6dc7693e9daf17c03a74183e5ab5cacbfd	2179
687836606e2b294cb8aa6eb71c6827564dcbf2f229eebfd9033c9222545d9bff75b086fa56d881dc63600ba0bd24a06ff83b4b988621c1e77e1cf22df6db8dd3	2182
d08051ce908afc6f24cf2c63b66ceb1c9f8648a506bd33a416051d8de79062c65f774a7bc8d658bd6d0f0967ef29ec5691bc31665abeb30c233762e68d7315ff	2183
2aa61a7ab7f29e2f25159af38779862289ea2512f80b3282d038dfb1abb9688f6ff1e364423b99d6cae24491d4322c9ffae1dbef51dbc95fd0c387d9fb7c0ae9	2184
a5b7bc7fca65b63a5dccc1a616db565f4866fe1d320e6a9daebbcbe78f0666cb003c9a002f1e486124512b0a3a90f531c3daa32c6b8ed375e9597ffd60e2dd9b	2185
a7989266db7835319e01b3ff69523253912c7aff7ed0ba6ac3073b5ce621200406c407a0f60f8259a90f46841f3a07ed37e186168e10891628095b2025c265b3	2186
cc3c8f47f58ad87ee72f4871727f0e89c0410e7f094fd780132b7765e9e18cbabc811283b2a47cd8bfb2d3a111473c2fe83209c7d49e650a2cb34ad454ac4b2e	2187
f56319c4e8a7206e409d226ce7b064d67f638031be0db1f76c71620bd47081faaaa3d0c6dc12cf49ed9658a5cd9b62301ba768b8be129938653f3a337e8218d6	2188
0aa8ce47cc9cfde448c47a13aa282d9cd63271256f49532ab1365482fcb968890cbf44d21269c277d487a88a09301c38bf46b9de7a8345c5c144e680c3083824	2189
5fdb9f31cd37ec95a121732f86de40dfc89b918fb87df8373d3f2a4515ebc0245a2b022e53da7755050a9257b4030b7c8c29b7819d94b460b60cd26425008c42	2190
42b97d8c1e0696444f9f2251e84a8833b8387c91ced492b26d81c1ea5563980f041e882d13a9451c7df3fabd47042b3aea6b6421551d3cf9192a03becffd8987	2191
316aba028a5d7b4ce33b566199573ab2aa727b6357efdf5165c087cf9aebd5603f267901e99fa437ec5c664a123348cef3253a5e6dc72a97ee8bd7bf3aa3e147	2192
ee715b9a5159b93d801735d4004397bac20a0a5f361dfa1eb98a67c822b677080c24c207f12ebd74b2748962f8653ca7c999c5e14f5b982964bd734939f4c8b3	2193
f0fbace1f5df0ca9d90647371697598e8dcd002c8e8428045526d7b72daf85c36dab1b903dc57a8896e18637b78f9629810b400b77aed650542098322add739a	2194
d3ff5946221a64195b7266d7685ca5170623d64bd383b7f357c2700a409927ebfa7f21d1751a6155d54f5061765264ce9bc6a6809d5348dffd41c558814837cf	2195

Table B.8.: Mozilla Common Voice speaker IDs alignment.

Dataset	Sex	Duration (hours)	Duration after removing silence (hours)	Training time (hours)	Quality	Note
2176	male	1.12	-	-	2	-
2179	male	1.02	-	-	2	-
2182	male	1.10	-	-	1	acoustic differences
2183	male	1.18	-	-	1	acoustic differences
2184	male	1.62	-	-	2	-
2185	male	2.59	-	-	1	acoustic problems
2186	male	2.73	-	-	2	best between male
2187	male	3.22	-	-	1-2	too quiet
2188	mix	2.25	-	-	0	multiple speakers
2189	female	2.19	-	-	2	best between female
2190	male	3.15	-	-	1	acoustic problems
2191	male	2.71	-	-	1-2	too quiet
2192	male	2.49	-	-	0	multiple speakers
2193	male	2.60	-	-	0	multiple speakers
2194	male and female	3.31	-	-	0	multiple speakers
2195	male and female	4.00	-	-	0	multiple speakers

Table B.9.: Mozilla Common Voice speech datasets.

C. Evaluation sentences

Text sentences, used for evaluation of a system, are found here.

C.1. English model

Test sentences for evaluation are divided into three groups (one group for each stage of evaluation):

- SUS (based on structure given in Gibbon et al. (1997) and NIT (2005)) and Harvard sentences (taken from IEEE (1969));
- Harvard sentences of different length;
- Text from open-source utterances of different TTS systems.

C.1.1. Stage 1

The SUS are:

1. The table walked through the blue truth.
2. The robust visitors grew in the flowering pool.
3. The strong way drank the day.
4. The mysterious bullets began the mythological browser.
5. Never draw the house and the fact.
6. Kindly bring the hope to the library.
7. How does the day love the bright word?
8. Why does the jazz hit the brown bar?
9. The place closed the fish that lived.
10. The dishonest audiences began the rosemary that flew away.

The Harvard sentences are:

1. Use a pencil to write the first draft.
2. The train brought our hero to the big town.
3. She did her best to help him.
4. The beauty of the view stunned the young boy.
5. Please wait outside of the house.

C.1.2. Stage 2

The Harvard sentences for stage 2 are:

1. One.
2. Hurry up!
3. Is it free?
4. A person thinking.
5. Christmas is coming.
6. A compromise disappears.
7. Fasten two pins on each side.
8. He ran half way to the hardware store.
9. Let it burn, it gives us warmth and comfort.
10. She wrote him a long letter, but he didn’t read it.
11. The square wooden crate was packed to be shipped.

12. The weight of the package was seen on the high scale.
13. It was hidden from sight by a mass of leaves and shrubs.
14. The bills were mailed promptly on the tenth of the month.
15. I currently have four windows open up and I don't know why.
16. A generous continuum of Amazon dot com is the conflicting worker.
17. If purple people eaters are real where do they find purple people to eat?
18. Italy is my favorite country, in fact, I plan to spend two weeks there next year.
19. She works two jobs to make ends meet, at least, that was her reason for not having time to join us.
20. If the easter bunny and the tooth fairy had babies, would they take your teeth and leave chocolate for you?

C.1.3. Stage 3

The sentences from Tacotron 2 (Shen et al., 2018) are:

1. Take a look at these pages for crooked creek drive.
2. There are several listings for gas station.
3. Here's the forecast for the next four days.
4. Here is some information about the Gospel of John.
5. His motives were more pragmatic and political.
6. She had three brothers and two sisters.
7. This work reflects a quest for lost identity, a recuperation of an unknown past.
8. He was being fitted for ruling the state, in the words of his biographer.

The sentences from DC-TTS (Tachibana et al., 2017) are:

1. The birch canoe slid on the smooth planks.
2. It's easy to tell the depth of a well.
3. The box was thrown beside the parked truck.
4. Four hours of steady work faced us.
5. Large size in stockings is hard to sell.
6. The boy was there when the sun rose.
7. A rod is used to catch pink salmon.
8. Kick the ball straight and follow through.
9. Help the woman get back to her feet.
10. A pot of tea helps to pass the evening.
11. The soft cushion broke the man's fall.
12. The salt breeze came across from the sea.

The sentences from Sample Efficient Adaptive Text-to-Speech (Chen et al., 2018) are:

1. Here are some pages for who sells Toms shoes.
2. Modern birds are classified as coelurosaurs by nearly all palaeontologists.
3. There were many editions of these works still being used in the 19th century.
4. The town is further intersected by numerous small canals with tree-bordered quays.

The sentences from Deep Voice 3 (Yamamoto et al., 2018) are:

1. Scientists at the CERN laboratory say they have discovered a new particle.
2. There's a way to measure the acute emotional intelligence that has never gone out of style.
3. President Trump met with other leaders at the Group of 20 conference.
4. The Senate's bill to repeal and replace the Affordable Care Act is now imperiled.
5. Generative adversarial network or variational auto-encoder.
6. The buses aren't the problem, they actually provide a solution.

C.2. German model

Test sentences for evaluation are divided into two groups (one group for each stage of evaluation):

- SUS (based on structure given in Gibbon et al. (1997)) and Marburg (taken from Brinkmann (1974)) sentences;
- Wenker (taken from Wenker (1880)) and Marburg sentences of different lengths.

C.2.1. Stage 1

The SUS are:

1. Böse Menschen fallen auf die Sonne.
2. Die katholischen Musicals schlafen im Gras.
3. Der treue Hund tadelt die nasse Ente.
4. Die unsichtbaren Würste haben das Präsidentenprojekt neu geschrieben.
5. Nehmt doch Leben zum Hafen!
6. Schaue immer mit einem Baum am Hinterkopf.
7. Wo erlaubt die Stärke die feine Wahl?
8. Wieso angelt der Sinn unter dem Fahrzeug?
9. Der Bauer grüßt die Ärztinnen, die lächeln.
10. Die naturalistischen Konzepte retten den Zombie, der ein Auto fährt.

The Marburg sentences are:

1. Wer weiß dort genau Bescheid?
2. Du darfst Dich wieder setzen.
3. Unser Haar braucht Pflege.
4. Es geht hier ums Prinzip.
5. Bitte öffnet doch gleich beide Türen!

C.2.2. Stage 2

The Wenker and Marburg sentences for stage 2 are:

1. Zwei.
2. Abendbrot.
3. Deine Uhr geht vor.
4. Wir spielen alle Tage.
5. Leider ist dies Haus teuer.
6. Gut Ding will Weile haben.
7. Dort muss jedes Auto bremsen.
8. Nicht jeder verträgt kaltes Bier.
9. Wir sind müde und haben Durst.
10. Ich habe das akustisch nicht verstanden.
11. Motoren brauchen Benzin, Öl und Wasser.
12. Verkehrsampeln leuchten grün, gelb und rot.
13. Schulkinder müssen Rechnen und Schreiben lernen.
14. Die Leute sind heute alle draußen auf dem Feld und mähen.
15. Ich verstehe euch nicht, ihr müsst ein bisschen lauter sprechen.
16. Es hört gleich auf zu schneien, dann wird das Wetter wieder besser.
17. Der Schnee ist diese Nacht liegen geblieben, aber heute morgen ist er geschmolzen.
18. Der gute alte Mann ist mit dem Pferd auf dem Eis eingebrochen und in das kalte Wasser gefallen.
19. Als wir gestern abend zurück kamen, da lagen die anderen schon im Bett und waren fest eingeschlafen.
20. Du bist noch nicht groß genug, um eine Flasche Wein allein auszutrinken, du musst erst noch größer werden.

Acronyms

ANN Artificial Neural Network. 5, 7–10, 18, 19, 25, 36
ARPA Advanced Research Projects Agency. 50
ASCII American Standard Code for Information Interchange. 50
ASR Automatic Speech Recognition. 12, 19

CEFR Common European Framework of Reference for Languages. 30, 34
CNN Convolutional Neural Network. 8–10, 12, 36
CPU Central Processing Unit. 3, 25

dB Decibel. 22, 50
DNN Deep Neural Network. 3, 5, 7, 15, 36

FFT Fast Fourier Transform. 51

GPU Graphics Processing Unit. 3, 25, 36, 41–44
GUI Graphical User Interface. 28

HMM Hidden Markov Model. 3, 15

ID Identifier. 22, 43, 45

LSTM Long Short-Term Memory. 10

MLP Multi-Layer Perceptron. 6, 7
MOS Mean Opinion Score. 27, 28, 30, 32, 34–36

NLP Natural Language Processing. 3, 8, 10, 30, 34

RAM Random-Access Memory. 41
RMSE Root Mean Squared Error. 27
RNN Recurrent Neural Network. 10, 40

SPSS Statistical Parametric Speech Synthesis. 15
STFT Short-Time Fourier Transform. 12
SUS Semantically Unpredictable Sentences. 27, 29–31, 34, 46, 48

TDNN Time-Delay Neural Network. 9
TTS Text-to-Speech. 3–5, 7, 8, 10, 12, 13, 15, 17–19, 22, 24, 25, 27, 28, 30, 32, 34–38, 41, 46

WAVE Waveform Audio File Format. 17, 37
WER Word Error Rate. 5, 27, 30–32, 34, 36

Glossary

Anglicism a word or phrase borrowed from English into a foreign language. 37

ARPAbet a set of phonetic transcription codes developed by the Advanced Research Projects Agency (ARPA). It represents phonemes and allophones of General American English with distinct sequences of ASCII characters. 14

autoregressive denoted to autoregression: a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step. 16

backend the part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data. 13

envelope referring to an oscillating signal: a smooth curve outlining its extremes. 51

frontend the part of a computer system or application with which the user interacts directly. 5, 13, 22, 25, 26

fundamental frequency the lowest (thus the loudest) frequency of a periodic waveform. 12, 15

glottal pulse a short burst of air, emerging from the lungs through the glottis, resulting in a periodic pulse train with an audible pitch, which is then passed through multiple filters (tongue, lips, etc.) to produce speech. 14

glottis the opening between the vocal folds (folds of tissue in the throat), which create sounds through opening and clothing as well as vibrating. 50

grapheme the smallest meaningful contrastive unit in a writing system. 16, 22

Harvard sentences a collection of sample phrases that are used for standardized testing of telecommunications, speech, and acoustics systems. They are phonetically balanced sentences that use specific phonemes at the same frequency they appear in English. 29, 46

Likert scale a scale used to represent people's attitudes to a topic, in which the individual is allowed to express how much they agree or disagree with a particular statement. 28

mel scale a perceptual scale of pitches judged by listeners to be equal in distance from one another, as it is heard by the human ear. The reference point between this scale and normal frequency measurement is defined by assigning a perceptual pitch of 1000 mels to a 1000 Hz tone, 40 dB above the listener's threshold. As frequency rises, increasingly large intervals are judged by listeners to produce equal pitch increments. 50

mel-scale log-magnitude spectrogram usually the same as mel-spectrogram, but with an explicit indication that the magnitude of the signal is transformed to log scale (usually to dB). 12, 16

mel-scale spectrogram the same as mel-spectrogram. 16

mel-spectrogram a spectrogram with the mel scale as its y-axis (see Figure 6.2). Generated as decomposition of the magnitude of the signal into its components, corresponding to the frequencies in the mel scale. 12, 17, 50

mini-syllabic words the shortest available words in their class. 27

overfitting the production of an analysis that corresponds too closely or exactly to a particular set of data, and may, therefore, fail to fit additional data or predict future observations reliably. 9, 17, 25, 26

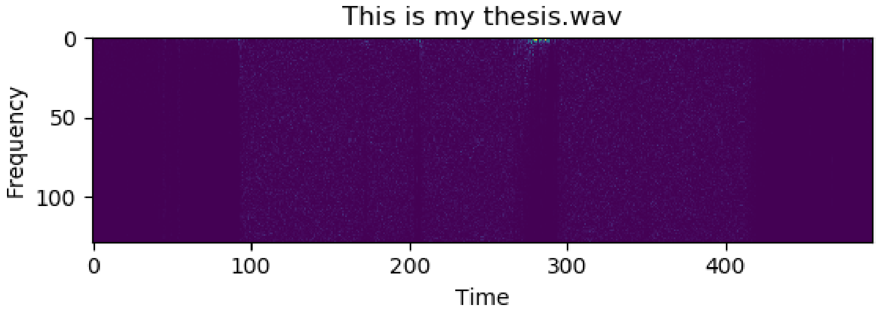


Figure 6.1.: Spectrogram for phrase: *"This is my thesis"*.

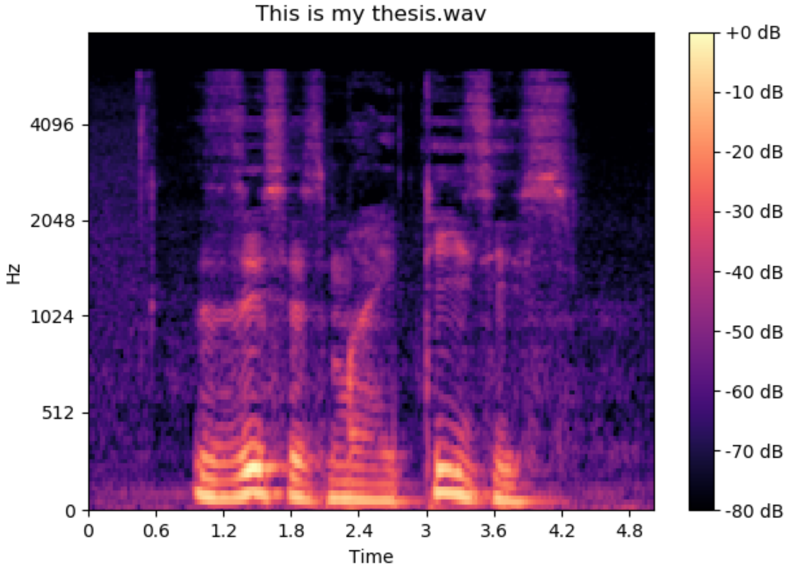


Figure 6.2.: Mel-spectrogram for phrase: *"This is my thesis"*.

phoneme any of the perceptually distinct units of sound in a specified language that distinguish one word from another. 9, 11, 12, 14, 16, 17, 22, 25

Q-word a function word used to ask a question, such as what, when, where, who, which, whom, whose, why, and how. The other names are interrogative word or question word. 29

redundancy in linguistics, refers to information that is expressed more than once. 27, 30, 34, 35

sample rate rate, with which sampling happens, herewith sampling is the reduction of a continuous-time signal to a discrete-time signal. 42, 44

spectral envelope the envelope curve of the amplitude spectrum (or audio power spectrum). 12

spectrogram a visual representation of the spectrum of frequencies of a signal as it varies with time (see Figure 6.1). Transformation of signal from time domain to frequency domain originates from FFT for each window. 12

Bibliography

Nagoya Institute of Technology (NIT) English sentences. `http://research.nii.ac.jp/src/en/NITECH-EN.html`, 2005. 46

CMU Pronouncing Dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`, 2014. 14

CMU Sphinx – Speech Recognition Toolkit. `https://sourceforge.net/projects/cmusphinx/`, 2019. 22

J. Allen, M. S. Hunnicutt, D. H. Klatt, R. C. Armstrong, and D. B. Pisoni. *From Text to Speech: The MITalk System*. Cambridge University Press, New York, NY, USA, 1987. ISBN 0-521-30641-8. 14

D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, art. arXiv:1409.0473, Sep 2014. 11

C. Benoît. An intelligibility test using semantically unpredictable sentences: towards the quantification of linguistic complexity. *Speech Communication*, 9(4):293 – 304, 1990. ISSN 0167-6393. doi: https://doi.org/10.1016/0167-6393(90)90005-T. URL `http://www.sciencedirect.com/science/article/pii/016763939090005T`. 27

C. Benoît, M. Grice, and V. Hazan. The SUS test: A method for the assessment of text-to-speech synthesis intelligibility using Semantically Unpredictable Sentences. *Speech Communication*, 18(4):381 – 392, 1996. ISSN 0167-6393. doi: https://doi.org/10.1016/0167-6393(96)00026-X. URL `http://www.sciencedirect.com/science/article/pii/016763939600026X`. 27

P. Birkholz. VocalTractLab: Towards high-quality articulatory speech synthesis. `http://www.vocaltractlab.de/index.php?page=background-articulatory-synthesis`, 2017. 14

C. M. Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006. ISBN 0-387-31073-8; 978-0-387-31073-2. 5, 15

A. Black and P. Taylor. CHATR: a generic speech synthesis system. In *COLING94*, pages 983–986, Kyoto, Japan, 1994. 14

K. Brinkmann. Die Neuaufnahme des Marburger Satzverständnistestes/The New Recording of the Marburg Sentence Intelligibility Test. *Zeitschrift für Hörgeräte-Akustik: Internationale Beiträge über Audiologie und deren Grenzgebiete/Journal of Audiological Technique: International Studies of Audiology and Related Fields*, 13(6):190–206, Nov. 1974. URL `https://www.uzh.ch/orl/dga-ev/publikationen/zfaudiologie/archiv/HGAk_1974_13-6_190-206_Original.pdf`. 29, 48

Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, C. Gulcehre, A. van den Oord, O. Vinyals, and N. de Freitas. Sample Efficient Adaptive Text-to-Speech. *arXiv e-prints*, art. arXiv:1809.10460, Sep 2018. 32, 47

J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-Based Models for Speech Recognition. *arXiv e-prints*, art. arXiv:1506.07503, Jun 2015. 11

H. Cryer and S. Home. Review of methods for evaluating synthetic speech. Technical Report 8, RNIB Centre for Accessible Information, 58-72 John Bright Street, Birmingham, UK, Feb. 2010. 27

A. K. Datta. *Epoch Synchronous Overlap Add (ESOLA) : A Concatenative Synthesis Procedure for Speech*. Signals and Communication Technology. Springer, Singapore, 2018. ISBN 9789811070167. URL `https://doi.org/10.1007/978-981-10-7016-7`. 14

A. Dertat. Applied Deep Learning - Part 4: Convolutional Neural Networks. `https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2`, Nov. 2017. 9, 10

G. Dreyfus, editor. *Neural networks: methodology and applications*. Springer, Berlin, 2005. ISBN 3-540-22980-9; 978-3-540-22980-3. 5

V. Dupras. num2words – Convert numbers to words in multiple languages. `https://github.com/savoirfairelinux/num2words`, 2019. 22

G. Fant. *Acoustic theory of speech production*. Mouton, Hague, The Netherlands, 1960. 14

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional Sequence to Sequence Learning. *arXiv e-prints*, art. arXiv:1705.03122, May 2017. 10, 11

D. Gibbon, R. Moore, and R. Winski, editors. *Handbook of Standards and Resources for Spoken Language Systems*, chapter Conclusion: summary of test. Mouton de Gruyter, Berlin and New York, May 1997. 29, 46, 48

D. W. Griffin and J. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32:236–243, Apr. 1984. doi: 10.1109/TASSP.1984.1164317. 12, 17

J. Grimshaw, T. Bione, and W. Cardoso. *Who's got talent? Comparing TTS systems for comprehensibility, naturalness, and intelligibility*, pages 83–88. EuroCALL Conference short papers. Research-publishing.net, 2018. ISBN 978-2-490057-22-1. URL `https://doi.org/10.14705/rpnet.2018.26.817`. 27

A. Hunt and A. Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *ICASSP 96*, pages 373–376, Atlanta, Georgia, 1996. 13

IEEE. IEEE Recommended Practice for Speech Quality Measurements. *IEEE Transactions on Audio and Electroacoustics*, 17(3):225–246, Sep. 1969. ISSN 0018-9278. doi: 10.1109/TAU.1969.1162058. URL `https://www.cs.columbia.edu/~hgs/audio/harvard.html`. 29, 46

K. Ito. The LJ Speech Dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017. 3, 42

J. Kominek and A. W. Black. CMU_ARCTIC databases for speech synthesis. `http://www.festvox.org/cmu_arctic/`, 2003. 42

R. Kruse. *Computational Intelligence: Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze*. Computational IntelligenceSpringerLink. Springer Vieweg, Wiesbaden, 2. aufl. 2015 edition, 2015. ISBN 9783658109042. URL `https://doi.org/10.1007/978-3-658-10904-2`. 5, 8

Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, 1(4):541–551, Dec. 1989. ISSN 0899-7667. doi: 10.1162/neco.1989.1.4.541. URL `http://dx.doi.org/10.1162/neco.1989.1.4.541`. 9

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324, 1998. 9, 10

B. McFee, V. Lostanlen, M. McVicar, A. Metsai, S. Balke, C. Thomé, C. Raffel, D. Lee, K. Lee, O. Nieto, J. Mason, F. Zalkow, D. Ellis, E. Battenberg, V. Morozov, R. Yamamoto, R. Bittner, K. Choi, J. Moore, Z. Wei, nullmightybofo, P. Friesch, F.-R. Stöter, D. Hereñú, Thassilo, T. Kim, M. Vollrath, A. Weiss, C. Carr, and ajweiss dd. librosa/librosa: 0.7.0, July 2019. URL `https://doi.org/10.5281/zenodo.3270922`. 22

M. Morise, F. Yokomori, and K. Ozawa. WORLD: A Vocoder-Based High-Quality Speech Synthesis System for Real-Time Applications. *IEICE Transactions on Information and Systems*, 99:1877–1884, 2016. doi: 10.1587/transinf.2015EDP7457. 12, 17

Mozilla. Mozilla Common Voice. `https://voice.mozilla.org/en/datasets`, 2017. 42, 44

J. Mullennix and S. Stern. *Computer Synthesized Speech Technologies: Tools for Aiding Impairment*, page 11. IGI Global, 1 edition, Jan. 2010. 3

H. Murthy, R. Sinha, A.G.Ramakrishanan, S. Agrawal, M. D. Kulkarni, S. Chandra, R. Doctor, S. Lata, T. Patil, A. Prakash, A. Kesarwani, R. Raheja, and N. Yadav. Text to Speech Testing Strategy. `http://tdil-dc.in/undertaking/article/449854TTS_Testing_Strategy_ver_2.1.pdf`, July 2014. 27, 28

J. Navarrete. The SoX of Silence. `https://digitalcardboard.com/blog/2009/08/25/the-sox-of-silence/`, Aug. 2009. URL `http://sox.sourceforge.net/`. 22

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. LibriSpeech: An ASR corpus based on public domain audio books. `http://www.openslr.org/12/`, 2015. 42

K. Park and T. Mulc. German Single speaker Speech Dataset. `https://www.kaggle.com/bryanpark/german-single-speaker-speech-dataset`, 2018. 44

W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller. Deep Voice 3: Scaling Text-to-Speech with Convolutional Sequence Learning. *arXiv e-prints*, art. arXiv:1710.07654v3, Feb 2018. 5, 7, 3, 12, 16, 40

K. S. Rao. *Source Modeling Techniques for Quality Enhancement in Statistical Parametric Speech Synthesis*. SpringerBriefs in Speech Technology: Studies in Speech Signal Processing, Natural Language Understanding, and Machine LearningSpringerLink. Springer International Publishing, Cham, 2019. ISBN 9783030027599. URL `https://doi.org/10.1007/978-3-030-02759-9`. 13, 15

Riverbank Computing. PyQt. `https://riverbankcomputing.com/software/pyqt/intro`, 2019. 28

F. Rosenblatt. The Perceptron, a Perceiving and Recognizing Automaton Project Para. 1957. 5

U. Saxena. Speech synthesis techniques using deep neural networks. `https://medium.com/@saxenauts/speech-synthesis-techniques-using-deep-neural-networks-38699e943861`, Oct. 2017. 14, 15

S. Sharma. Activation functions in neural networks. `https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6`, Sept. 2017a. 6

S. Sharma. What the hell is perceptron? `https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53`, Sept. 2017b. 5

J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerry-Ryan, R. A. Saurous, Y. Agiomyrgiannakis, and Y. Wu. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. *arXiv e-prints*, art. arXiv:1712.05884v2, Feb 2018. 5, 7, 4, 8, 32, 47

J. O. Smith. *Physical Audio Signal Processing*. `http://ccrma.stanford.edu/~jos/pasp/`, accessed 08/01/2019. online book, 2010 edition. 14

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. *arXiv e-prints*, art. arXiv:1409.3215, Sep 2014. 10, 11

H. Tachibana, K. Uenoyama, and S. Aihara. Efficiently Trainable Text-to-Speech System Based on Deep Convolutional Networks with Guided Attention. *arXiv e-prints*, art. arXiv:1710.08969, Oct 2017. 32, 47

P. Taylor, A. Black, and R. Caley. The architecture of the Festival Speech Synthesis System. In *3rd ESCA Workshop on Speech Synthesis*, pages 147–151, Jenolan Caves, Australia, 1998. 14

N. Tits, K. El Haddad, and T. Dutoit. Exploring Transfer Learning for Low Resource Emotional TTS. *arXiv e-prints*, art. arXiv:1901.04276, Jan 2019. 15

I. Titze. *Principles of Voice Production*. Prentice Hall, 1994. ISBN 978-0-13-717893-3. 14

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A Generative Model for Raw Audio. *arXiv e-prints*, art. arXiv:1609.03499, Sep 2016. 12, 17

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention Is All You Need. *arXiv e-prints*, art. arXiv:1706.03762, Jun 2017. 12

W. von Kempelen. *Mechanismus der menschlichen Sprache nebst der Beschreibung seiner sprechenden Maschine*. Degen, 1 edition, 1791. 3

VoxForge. VoxForge Speech Corpora. `http://www.voxforge.org/home/Downloads`, 2006. 42, 44

A. Waibel, T. Hanazawa, G. E. Hinton, K. Shikano, and K. Lang. Phoneme Recognition Using Time-Delay Neural Networks. Technical Report TR-I-0006, Advanced Telecommunications Research Institute, International Interpreting Telephony Research Laboratories, Japan, Oct. 1987. 9

A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339, March 1989. doi: 10.1109/29.21701. 9

G. Wenker. Deutscher Sprachatlas (DSA). `https://wolfgang-naeser-marburg.lima-city.de/htm/wenker.htm`, 1880. 29, 48

R. Yamamoto, H. Kim, cclauss, homink, amilamad, T. Sereda, S. Fischer, Raúl, lzala, and K. Mametani. PyTorch implementation of convolutional neural networks-based text-to-speech synthesis models, Oct. 2018. URL `https://doi.org/10.5281/zenodo.1472613`. 3, 32, 47

Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran. Learning to Speak Fluently in a Foreign Language: Multilingual Speech Synthesis and Cross-Language Voice Cloning. *arXiv e-prints*, art. arXiv:1907.04448, Jul 2019. 15