Project Thesis 1: Towards Predicting Speech with Sequence Learners*

Pascal Sager
ZHAW School of Engineering
Winterthur, Switzerland
sagerpa1@students.zhaw.ch

***Abstract*—Current research in the area of speech data analysis with deep learning focuses mainly on speech recognition, speech segmentation, speaker recognition, speech classification and speech synthesis. The prediction of frames of speech is a less studied area. In this thesis, a neural network based on Gated Recurrent Units (GRUs) is proposed to predict the subsequent frames of given Mel-spectrograms. The model is trained on the TIMIT dataset to predict sequences up to one word in length. It is found that an input length of approximately 2.5 words performs best for the proposed network. Analyzing the mean absolute error shows that frames closer in time to the given sequence require less input data than frames that are further away. Moreover, it is observed that certain phoneme types such as closures, pauses, fricatives and affricates are predicted worse than other phonemes. The main issue of regressive models to predict frames of speech is that they tend to predict ranges where the formants might lie instead of specific positions. As a result, the predicted Mel-spectograms look blurry and have missing details. Various well-known distance metrics such as mean absolute error, mean squared error, or soft dynamic time warping could not alleviate this problem. However, it is observed that the system exhibits different characteristics depending on the metric used as loss function. Based on the gained experience, different measures such as learning a proper distance metric or alternative architectures are proposed to improve the overall prediction quality.**

***Index Terms*—speech prediction, audio processing, deep learning**

I. Introduction

In our everyday lives we encounter a variety of audio signals such as human speech, music, animal voices or sounds from human activity such as cars and machinery. Given the prevalence of sounds, it is no surprise that there exist a vast number of use cases for audio processing. Many of these audio processing applications such as speaker recognition [1], speech recognition [2], audio separation [3], audio segmentation [4], audio classification [5] or text to speech conversion [6] are scientifically well-studied areas. However, the prediction of speech has been examined less.

In the field of text data processing, the prediction of subsequent words has been widely researched. Such systems are used, for example, for word correction and spell check systems [7] or to compute vector representations of words [8]. While systems for the prediction of written language are well studied, the prediction of spoken language remains relatively under-examined.

Processing audio signals is complex and characterised by various challenges. For example, speech processing systems need to be resistant to background noise but still tolerant to slight variations in the speed and pitch of a signal [9]. For this reason, researchers have been working on more reliable systems for decades. Earlier systems used methods such as Vector Quantization (VQ) [10], Hidden Markov Models (HMM) [11] or Gaussian Mixture Models (GMM) [12]. Many systems that have been developed in recent years use the powerful feature extraction capabilities of deep neural networks (DNNs) [13]–[15]. More precisely, mainly convolutional neural networks (CNNs) [16], recurrent neural networks (RNNs) [17] or Transformer [18] are used nowadays to process audio data.

In this project thesis, the prediction of frames of Mel-spectrograms [19] is investigated using deep learning methods. Thereby, the prediction is done without conversion to text data (i.e. not combining existing speech recognition systems, text prediction systems and speech synthesis models). The task of predicting audio data is particularly interesting for two reasons: First, such systems can be used in practice, for example to optimize audio interfaces by completing truncated signals. Second, such models can generate features from Mel-spectrograms [20] which can be used for other downstream tasks such as speaker classification. However, this project thesis focuses on the accurate prediction of frames of speech rather than feature generation for other tasks.

II. Related Work

To the best of the author's knowledge, no work has been published that investigates the prediction of sequences of Mel-spectrograms. Nevertheless, auto-regressive models for

* This project thesis was written by Pascal Sager as part of the "Master of Science in Engineering with Specialisation in Data Science" program at the Zurich University of Applied Sciences. This thesis was supervised by Prof. Dr. Thilo Stadelmann. The code is publicly available on Github https://github.com/sagerpascal/speech-prediction.

generating speech representations as well as speech synthesis models are considered related. Auto-regressive models for generating speech representations are trained to predict single frames of Mel-spectrograms and subsequently used to generate representation vectors for other tasks. Speech synthesis models, on the other hand, often generate Mel-spectrograms based on text data and synthesize them into waveforms.

A. Auto-Regressive Models for Speech Representation Learning

In the field of text data processing, auto-regressive models are often used for unsupervised pre-training [21]. This concept applied to big data has led to the development of very advanced and well-known models in recent years such as GPT v1-v3 [22]–[24], Transformer-XL [25] or Reformer [26].
Recently, this concept has been applied to audio data as well [20], [27]. Thereby, mainly Mel-spectrograms are used and based on given frames a subsequent frame is predicted. This pre-training takes place on large datasets such as LibriSpeech [28]. The goal of the pre-training is to learn speech representations that can be used for different downstream tasks across different datasets. Depending on the task, speech information from different layers of the model are extracted. In particular, the lower layers capture more information about the speakers, while the upper layers capture more phonetic content.
However, instead of focusing on the accurate prediction of frames, these models aim to generate good representations of speech. Moreover, these systems are only used to predict single frames with a specific offset from the given frames. These systems are not able to predict multiple subsequent frames based on a given sequence and can therefore not be used for the prediction of speech.

B. Speech Synthesis

Speech synthesis models, also called text-to-speech (TTS) models, generate waveforms from text data. Many prominent methods such as Tacotron [29], Tacotron 2 [30] or FastSpeech [31] generate Mel-spectrograms based on written text and then synthesize raw waveforms from the Mel-spectrograms. The models described use two different strategies to convert text to Mel-spectrograms: Tacotron and Tacotron 2 use an end-to-end approach based on a recurrent sequence-to-sequence feature prediction network. FastSpeech, on the other hand, trains an additional phoneme duration prediction module and argues that this two-staged approach increases robustness and contrability of the speaking rate. These networks are related because even tough they have a different type of input (i.e. text), they also generate Mel-spectrograms. Thus, similar principles can be applied for the generation process.
Additionally, speech synthesis models are relevant when predicted Mel-specotrgrams are re-synthesized into waveforms. The Tacotron model converts the Mel-spectrogram into a linear-scale spectrogram, using a CBHG module consisting of 1D convolutional filters, highway networks [32] and bidirectional GRUs. Afterwards, the Griffin-Lim [33] algorithm is used to reconstruct the signal. Other approaches use mainly WaveNet [34] as vocoder or slight variations of this network such as WaveGlow [35] or Parallel WaveGAN [36] for the synthesis.
WaveNet is a generative model for raw audio waveforms and was trained by predicting the next value of a waveform signal using dilated causal convolutional layers [34]. However, the authors of WaveNet state that the model is not feasible for speech prediction due to the lack of long range coherence. This is because the waveforms consists of $8'000$-$16'000$ measurements per second and therefore a very large receptive field would be needed.

III. Concept

In this project thesis, the audio files from the TIMIT [37] speech corpus are used. This dataset includes recordings of 630 speakers of eight major dialects of American English reading ten out of 2342 different phonetically rich sentences. The TIMIT dataset was chosen mainly due to its cleanness as well as the time-aligned phonetic transcriptions, which are useful for evaluation purposes.
The recorded audio data in this corpus is the measured air pressure per time and is converted to a digital signal via sampling [38] with a sampling rate of 16kHz. This digital speech signal has one dimension but contains information about the linguistic content, background noise, as well as information about the speaker (e.g. gender, origin, emotional state etc.). In order to better separate these different kinds of information, the speech signal is transformed into the frequency domain.
The transformation from the time domain to the frequency domain is done using the Fast Fourier Transformation (FFT) [39], which requires the signal to be static. Therefore, the quasi-stationary speech signal is split into frames of 25ms length using the Hann window function [40] with a window-size of 400 sample points. During this short period, the statistical parameters of the signal are relatively stable and the FFT can be applied.
With the Fourier transformation, the time domain of a signal is traded for the frequency domain. The result of the FFT is a spectrum that represents the energy per frequency. However, this spectrum only contains the energy per frequency for a single frame (i.e. one sequence extracted with the Hann window). Several spectrums are calculated by shifting the Hann window by 50 percent of its size further on the time axis. Thereby, the FFT is computed on overlapping windowed segments of the signal, and the resulting spectrums are stacked on each other. Through the simultaneous capture of the time-frequency plane of a speech signal, the so called spectrogram is calculated. It represents the energy per frequency over the frames.
Humans do not perceive frequencies on a linear scale and are better in distinguishing lower frequencies than higher frequencies. Therefore, Volkman et al. [41] proposed the Mel scale which scales the signal such that equal distances in pitch sound equally distant to the listener. This scale also helps to interpret the spectrogram and was therefore applied on

the spectrogram. Spectrograms that use this scale are called Mel-spectrograms and show the amplitude per frequency over time.

Figure 1 depicts the entire concept of the application. It shows the described transformation of the raw signal (i) into a Mel-spectrogram (iv) as *signal transformation* (iii). Before the described transformation, data augmentation is optionally applied (ii). Data augmentation helps to obtain models that generalise better and is described in more detail in section IV-A.

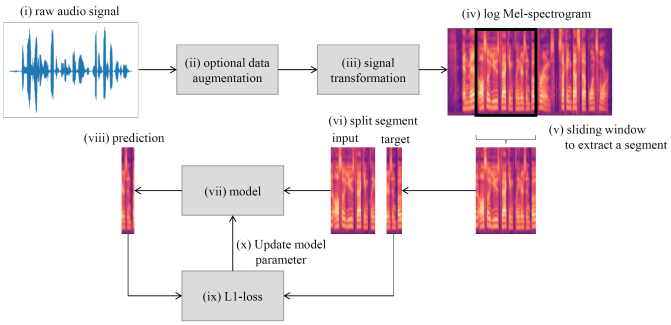

Fig. 1. The concept of this work to predict sequences of Mel-spectrograms: First, the raw audio signal is loaded (i) and optionally data augmentation (ii) is applied. Then the audio signal is transformed (iii) to a Mel-spectrogram (iv). A sliding window (v) is used to extract a segment, which is then split into input and target (vi). A model (vii) is trained to predict the target sequence (viii) based on the input sequence. Finally, the model parameters are updated (x) by comparing the target sequence with the prediction using the Mean Absolute Error (MAE) loss function (ix).

After the Mel-spectrogram of an audio signal is calculated, a sliding window is shifted over the time axis of the Mel-spectrogram (v). The sliding window is used to extract segments with a fixed length. The extracted segment is then split into two sub-segments (vi). The chronological first segment contains n frames and is fed as an input into the model. The chronological second segment consisting of k frames is used as the target of the prediction task. The network is trained to predict the unseen segment consisting of k frames based on the given segment with n frames.

The described sliding-window approach is a case of self-supervised learning, where the targets are derived from the input data. Self-supervised learning not only eliminates labelling costs, but also prevents label corruption and makes it straightforward to add new data [42].

In this work, the window for extracting segments has been shifted by one frame on the time axis. This implies that target vectors y_{t_1} are reused in a subsequent training sample as input vectors x_{t_2}. The reuse of target vectors is inspired by similar training methods from the field of text processing. For example, training methods of word2vec models [8] such as c-bow or skip-gram also predict the context within a window and reuse the target tokens as input tokens.

The definition of the number of given frames n and the number of frames to predict k has a significant impact on the results. In this work, the models were trained to predict one word. This means that the parameter k was fixed to a specific number of frames. Gráf [43] measured that a native English speaker speaks 196 words per minute on average. With this assumption it can be calculated that one word corresponds to 306ms as shown in equation 1.

$$t_{1\mathrm{w}} = \frac{1\mathrm{w} \cdot 60\mathrm{s}}{196\mathrm{wpm}} \approx 0.306\mathrm{s} \tag{1}$$

By a given sample rate of $f_s = 16\mathrm{kHz}$, a Hann window size of $h_l = 400$ frames and a window shift of $h_s = 200$ frames, 306ms corresponds to 24.5 frames as shown in equation 2. Therefore, the parameter k was set to $k = 25$ and thus the model was trained to predict approximately one word of speech.

$$k_{1\mathrm{w}} = \frac{t_{1\mathrm{w}} \cdot f_s}{\frac{h_l}{h_l/h_s}} = \frac{0.306\mathrm{s} \cdot 16000\mathrm{s}^{-1}}{\frac{400}{2}} \approx 24.5 \tag{2}$$

The second parameter n which determines how many frames are used as an input to the model was treated as a hyper-parameter. If more frames are fed into the model (i.e. n is larger), the model has more information available. Theoretically, this allows the model to extract more speaker-dependent features as well as more information about the context than from shorter sequences. However, longer sequences also have disadvantages. For example, models based on RNNs process the data sequentially. For n given frames, this leads to n recurrent steps which cannot be parallelized. Therefore, the number of given frames n has a significant impact on the performance of the entire model. Another disadvantage of longer input sequences is that less segments can be extracted from the Mel-spectograms which results in fewer training samples.

Auto-regressive models for text typically calculate the probability of a token at time t, given the previous tokens $(x_{t-n}, x_{t-(n-1)}, ..., x_{t-1})$. Therefore, they usually use a Softmax layer at the end of the network to estimate the probability distribution over the tokens [24], [25], [44]. However, for speech data, each token t_k corresponds to a frame rather than a written word. Since the set of target tokens for speech data is infinite, the Softmax layer is replaced with a regression layer. Consequently, the model directly predicts the k subsequent target frames for n given frames and does not calculate a probability over all existing frames.

The model is optimized by minimizing the L1 loss between the prediction and the ground truth, as it is done for many speech synthesis models [29], [45]. Different loss functions are examined in more detail in section V-A.

IV. Implementation

The Mel-spectrograms are calculated using 80 Mel-filterbanks. After applying the window function to extract a segment and splitting the segment into the two sub-segments of length n and length k, two vectors of the size $[80 \times n]$ and $[80 \times k]$ are obtained. The first vector consisting of n frames

is used as an input for the model and the second vector with k frames as a target.
As shown in figure 2, the model consists of a pre-net, a GRU-net and a post-net. This architecture is an extended version from Ref. [20] with additional post-processing.
First, the input vector is fed into the model's pre-net. The pre-net consists of 5 similar blocks, each containing a fully connected layer, followed by a ReLU activation, a dropout layer [46] and layer normalization [47]. This pre-net is used to extract features from the Mel-spectrogram. Thereby, the first fully connected layer increases the feature dimensionality per frame from 80 to 512. All the subsequent layers keep the dimensionality of 512 and thus the pre-net calculates latent representations with a dimensionality of $[512 \times n]$.

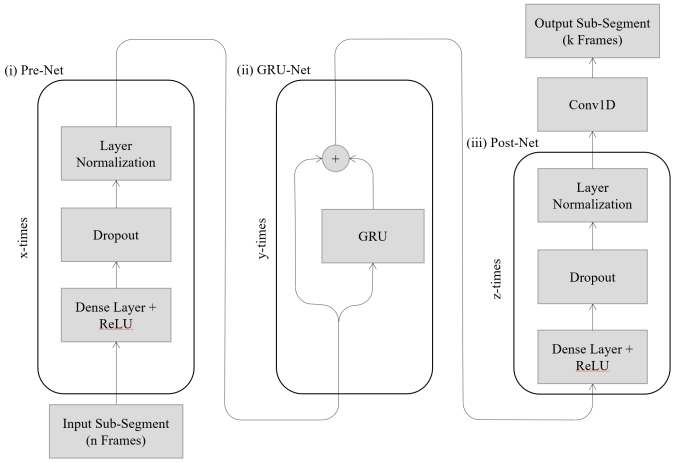

Fig. 2. The input sequence is first fed into a pre-net (i), which consists of 5 identical blocks, each containing a fully connected layer, followed by a dropout layer and layer normalization. The extracted latent representations are then fed into 4 GRUs (ii) with residual connections around it. Finally, the post-net (iii) maps the latent representations to the target size, and the final convolutional layer reduces the number of channels.

The GRU-net processes the generated latent representations. This sub-network consists of 4 Gated Recurrent Units with residual connections [48] around each unit. The residual connections allow the gradient to flow directly to the pre-net during backpropagation. This improves convergence by addressing well-known issues of RNNs such as vanishing or exploding gradients [49]. In addition to residual connections, gradient-clipping with a clip coefficient of $c = 1$ was used to further mitigate these issues.
Finally, a post-net is used to map the obtained sequence to the target number of frames $[512 \times k]$. The post-net consists of 3 similar modules, each containing a fully connected layer, followed by a ReLU activation, a dropout layer and layer normalization. Afterwards, a convolutional layer reduces the feature dimensionality per frame from 512 to 80 to obtain the target dimensionality of $[80 \times k]$.
The model was trained with the Adam optimizer [50]. Thereby, the learning rate was set to $\alpha = 8 \cdot 10^{-5}$, the weight decay to $d_w = 1 \cdot 10^{-4}$ and a mini-batch size of $b = 32$ was used.
In addition to the model proposed in this section, other architectures based on CNNs and sequence-to-sequence modeling were also evaluated. However, these architectures have performed worse and are therefore only described in the appendix in chapter A to provide insights for eventual follow-up work.

A. Data Augmentation

In order to achieve better generalization, data augmentation was used. The augmentation was directly applied to the raw signal and not to the Mel-spectrogram. It is important that the augmentation does not disrupt the raw signal excessively, otherwise the phonemes would not be clearly identifiable in the Mel-spectrogram and the performance would drop.
In this work, only a resampling function and an amplification method were used. The resampling augmentation method uses a high-quality implementation with a Kaiser window [51] for band-limited sinc interpolation. It was used with a probability of $p_{\text{resample}} = 0.75$ and resampled the original signal by a random factor between 0.7 and 1.3.
In addition to resampling, amplification of individual segments within the entire sequence was applied with a probability of $p_{\text{ampl}} = 0.75$. This augmentation method amplified or de-amplified random sub-sequences by a random factor between 0.8 and 1.2. Since the TIMIT corpus is relatively small, data augmentation is considered necessary to achieve good results on the test dataset.

B. Pre-Training

Pre-training was conducted for 10 epochs on the "train-clean-360" subset of the LibriSpeech corpus. This subset contains 360 hours of read English audio books [28]. During pre-training, the model learned general aspects of speech. After pre-training, the models were fine-tuned on the TIMIT corpus. The models with pre-training not only learned faster, but also generalised better. This indicates that some general features can be learned and transferred to other datasets with different speakers.

V. Preliminary Experiments

A. Loss Function

The loss function has a significant impact on the characteristics of the prediction. In the field of speech synthesis, the L1 loss is often used to optimize the model [29], [45]. The same applies for models which reduce noise in Mel-spectrograms [52]. Since the loss function in the domain of frames of speech prediction has not been investigated before, the following loss functions were evaluated in a preliminary experiment:

- *L1-Loss*: Mean Absolute Error (MAE)
- *L2-Loss*: Mean Square Error (MSE)
- *Soft-DTW L1-Loss*: Soft dynamic time warping with the MAE as distance metric
- *Soft-DTW L2-Loss*: Soft dynamic time warping with the MSE as distance metric
- *Adaptive Robust Loss Function*: A generalization of different loss functions which allows an automatic adaption of the robustness during training

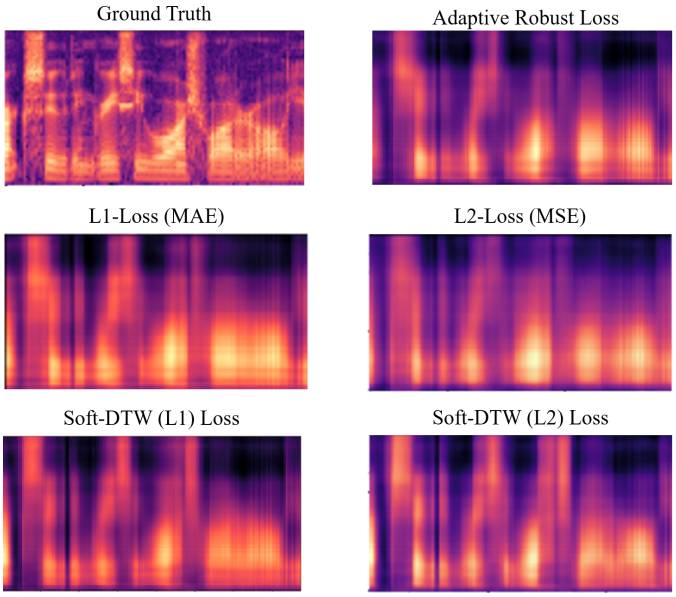

Fig. 3. The ground truth of a randomly chosen sentence and five corresponding predictions from models trained with a different loss function. In order to compare the whole sentence and not only a short sequence, several predictions were concatenated.

The L1 loss calculates the mean absolute error between the ground truth and the prediction in order to update the model. The L2 loss, on the other hand, gives more importance to large deviations by calculating the mean square error between the ground truth and the prediction.
A concern of these two loss functions is their limited robustness to shifts on the time axis [53]. For example, a high loss could occur if the prediction is good, but not well aligned on the time axis. To address this problem, soft dynamic time warping (soft-DTW) [54] was used for the L1 loss as well as for the L2 loss. Soft-DTW is based upon dynamic time warping (DTW) [53]. In general, DTW can compare vectors with different length in time and is robust to shifts or dilatations across the time dimension. Compared to DTW, soft-DTW computes the soft-minimum of all alignment costs. By doing so, the method becomes differentiable and can be used to optimize the model.
Furthermore, experiments with the adaptive robust loss function [55] were conducted. This loss was used to add more robustness to the model, i.e. that the model is less influenced by outliers than by inliers [56]. The adaptive robust loss function is a generalization of different loss functions with different robustness properties. By analyzing the gradients it automatically determines how robust the loss should be and adjusts the function accordingly without any manual parameter tuning.
Each of these loss functions were used to optimize a model. Figure 3 shows a representative sentence predicted by models trained with different loss functions. At a first glance, the predictions from these models look similar. However, a closer look reveals different characteristics.

Models trained with the adaptive robust loss or the L2 loss predict smaller amplitudes for the upper frequencies (i.e. the upper part of the Mel-spectrogram is less pronounced). The L1 loss and the two Soft-DTW versions, on the other hand, pronounce these upper frequencies stronger. The fact that the soft-DTW version of the L2 loss pronounces these higher frequencies more than the L2 loss indicates that dynamic time warping may be helpful for better predicting these upper frequencies.
It is also observable that the predictions of the L1 loss appear overall smoother and the phonemes are less separated. The adaptive robust loss and the L2 loss lead to predictions with more separated phonemes, while the two Soft-DTW losses show partly sharp transitions along the time axis.
However, the actual goal is the prediction of speech. Therefore, the acoustic perception is particularly important besides the visual evaluation of the Mel-spectrograms. For this purpose, the Mel-spectrograms were re-synthesized to waveforms using the Griffin Lim reconstruction algorithm [33]. Since the predicted Mel-spectrograms contain a lot of noise, perception experiments with a test group were not feasible and therefore not conducted. From the author's perception, the prediction of the model trained with the L1 loss sounded best.
The loss functions are further analyzed in the appendix in chapter B. By comparing time-shifted predictions with the average of the input, it is shown that the L1 loss is more robust than the L2 loss for predictions which are not well aligned on the time axis. In addition, it is found that the two soft-DTW versions are not well suited for predicting very short sequences. Consequently, the MAE was used as loss function for the project thesis but should be reconsidered in future work.

VI. Results

In all conducted experiments, the standard TIMIT training and test split [37] was used. For hyper-parameter tuning, the training dataset was subdivided and 10 percent of it was used as a validation dataset. After optimising the hyper-parameters, the model was retrained on the entire training dataset including the validation set. The test dataset was only revealed after tuning the hyper-parameters to evaluate the model. Thus, no information from the test split was incorporated into the training process or the optimization of the hyper-parameters.

Overall, the predicted sequences look blurry and some details are missing as shown in figure 4. For example, the formants are not clearly separated, and often only the outline of the predicted phonemes is identifiable but not the inner structure. This effect was observed for all tested models independent of the loss function. One reason for this behavior could be the fact that the formant frequency, the speaking rate, and the pronunciation are speaker-dependent [57]. For different speakers, this leads to different positions of the formants on the time-axis and on the frequency-axis of the Mel-spectrogram, even for identical sentences and words. This makes the prediction task for the model difficult. By using

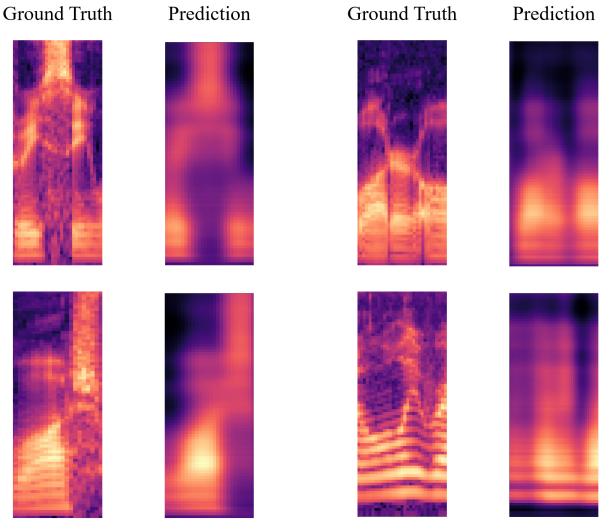

Fig. 4. Four random examples of predicted frames. The ground truth is shown on the left and the prediction of the model on the right.

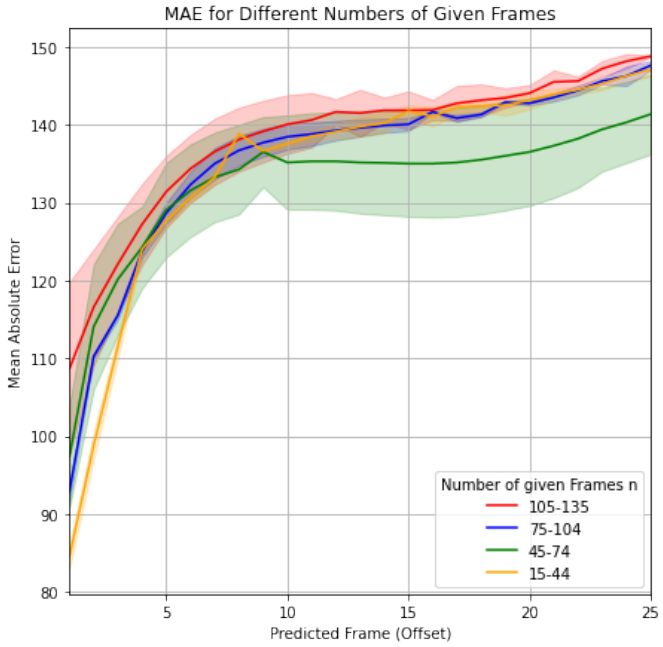

Fig. 5. The Mean Absolute Error (MAE) per predicted frame for various numbers of given frames. The y-axis shows the MAE, the x-axis show the predicted frame (e.g. 3 means the error for the 3rd frame) and the lines represent the error for different numbers of given frames.

simple distance metrics such as the L1 loss or the L2 loss, the model tends to predict a "range" in which the formants could lie, instead of predicting specific locations. Predicting such ranges instead of exact positions leads to the blurred looking predictions.

A. Number of Frames

While the number of frames to predict was fixed, the number of input frames n was treated as a hyper-parameter. Tuning this parameter is a trade-off between providing more information to the model (i.e. larger n) and having more training samples available (i.e. smaller n). Different numbers of frames were fed into the model and the MAE of each of the 25 predicted frames was calculated. Figure 5 shows the result.

The prediction accuracy of the first three frames was higher when only a few frames (n in the range of 15-44 frames) were fed into the model. This indicates that only a couple of frames are sufficient to predict the immediately following frames. At the same time, the first predicted frames benefited from having more training data. This suggests that the frames which are closer in time to the given data are based on more local information (i.e. more dependent on the immediately preceding frames).

Predicting frames that are further away in time from the given sequence requires more input data. For example, the prediction of the sequence that is 10-25 frames away from the given data achieved better results when more frames were given as an input (n in the range of 45 and 74 frames). This suggests that the prediction of frames further away in time rely on more global information. They benefit from more data being fed into the model even though if this leads to fewer training samples. The lowest MAE over all frames was obtained when approximately 60 frames were fed into the model. Given the assumption in equation 1 and the parameters from equation 2, 60 frames corresponds to ≈ 2.5 words as shown in equation 3.

This means that the best result to predict a word was obtained when 2.5 words were used as an input.

$$n_{60\text{fr}} = \frac{60 \cdot h_s}{f_s \cdot t_{1\text{w}}} = \frac{60 \cdot 200}{16000\text{s}^{-1} \cdot 0.306\text{s}} \approx 2.45 \qquad (3)$$

Because the frames of a Mel-spectogram evolve slowly over time, the frames that are closer to the input sequence have a lower MAE than the frames that are further away in time. The first predicted frames are more similar to the last given frames and consequently easier to predict. In addition, the uncertainty increases for frames that are further away in time. Figure 6 shows the predicted Mel-spectrograms with a fixed offset. For a given sequence, 25 frames were predicted, but only the frame at the position "offset" was kept. Afterwards, the sliding window was shifted forward by one frame on the time axis and the process was repeated. Finally, all kept frames were stacked and thus the resulting Mel-spectrogram shows the prediction with a specific offset from the given data.

The plots show that the prediction with an offset of one frame contains more detail. However, when frames that lay further in time are predicted, the predictions are blurrier. Consequently, the accuracy per predicted frame can not only be measured in numbers using the mean absolute error, but is indeed visible in the plotted Mel-spectrograms.

B. Error per Phoneme and Error per Speaker

The TIMIT dataset provides various additional information about the audio sequences. For example, time-aligned phonetic transcriptions and additional information about the speakers are included. The phonetic transcriptions were used

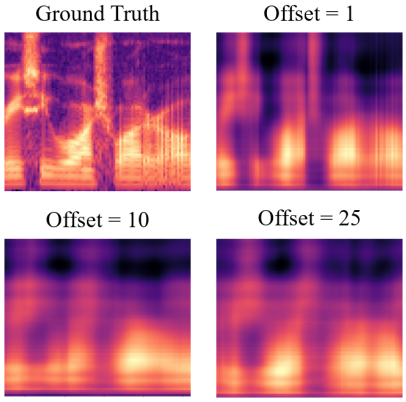

Fig. 6. The plot in the upper left corner shows the Mel-spectrogram of a randomly chosen sentence. For the other plots, multiple predictions were made and only the frame at the position "offset" was kept. Afterwards, all kept frames were concatenated. Thus, only the predicted frames that are a specific number of frames away from the given data are shown.

to evaluate the prediction accuracy per phoneme category. A detailed evaluation is included in the appendix in chapter C. The analysis of the prediction per phoneme type has shown that especially pause duration and closures of the lips are incorrectly predicted. This could be due to the facts that it is difficult in general to estimate transitions near a closure of the lips [58] and the pause durations is highly speaker dependent [59]. The model also has more difficulty predicting affricates and fricatives than other phoneme types. A reason for this could be that affricates and fricatives are generally difficult to determine, as their frequencies contain a random component by definition.

Chapter D of the appendix examines the prediction error per speaker. This evaluation shows that systems to predict frames of speech could have issues regarding fairness. For example, frames of speech spoken by individuals with a lower level of education or an African-American origin are predicted worse than frames spoken by individuals with a higher level of education or an European-American origin. This demonstrates that measures must be taken to ensure that such systems do not discriminate against groups of individuals based on attributes such as race, gender or education.

C. Perception of the Results

Some of the results can be found on https://sagerpascal.github.io/speech-prediction/results.html. The predictions are noisy which is reflected in the blurred plots of the predicted Mel-spectrograms as well as the re-synthesised waveforms. Nevertheless, most of the predicted words can be identified acoustically. This suggests that DNNs are able to predict speech.

D. Predicting Longer Sequences Using a Seed

So far, only $k = 25$ frames were predicted. Nevertheless, the number of frames to predict can be increased. However, by increasing this parameter also the error increases, because the further away a predicted frame is from the given frames, the less accurate the prediction becomes. Moreover, the number of predicted frames is always limited by an upper bound (i.e. the parameter k). Another approach to predict sequences of arbitrary length is to reuse the output of the model as an input. Thereby, an initial sequence (a.k.a. a seed) is fed into the model. The model then predicts the subsequent frames of this seed. By reusing this prediction as input, the model can theoretically predict the subsequent frames of the previous prediction. If this process is repeated continuously, sequences of any length can be predicted.

However, in this approach, the model faces the challenge that if a prediction contains noise and is reused as input, the next prediction must be made based on noisy data. Consequently, the prediction task becomes more difficult.

Another issue is that usually fewer frames are predicted than are needed as an input. Consequently, only a part of the input can be replaced by the prediction. Therefore, the chronologically oldest frames of the input are removed and the predicted frames are appended to the remaining input sequence. Hence, the input is gradually replaced by the predictions.

A model trained according the principle described in chapter III was not able to predict longer sequences. As soon as noisy predictions were fed into the model, the output became a constant vector.

Various measures were taken to counteract this behaviour. Chapter VI-A shows that the frames that are closer in time to the given data have a much smaller prediction error than the frames that are further away. Therefore, less frames were predicted and used as an input. By doing so, the output and therefore also the input becomes more accurate and the noise in the system is reduced.

As a second measure, the principle of reusing the output as an input has already been applied during training. Thereby, the model is explicitly trained and optimized on this specific task.

As a final measure, the L1 loss was replaced by a weighted L1 loss as it is done in the field of text data prediction for longer sequences [8]. Predicting frames further away from the seed becomes more challenging because of the accumulated noise. Therefore, a smaller weight was assigned to these frames so that they have less influence on the overall loss.

All these measures have contributed to improve the predictions. The model is in some cases able to predict longer sequences if a seed from the training dataset is used. However, with seeds from the test dataset, multiple words are only predicted in a few cases. Often the model collapses and predicts a constant vector for a longer time. In some cases, the model can recover and starts again to predict meaningful words.

VII. Future Work

The main problem with the system presented in this work is that the predictions are blurred. The model rather learns to predict a range where the formants of the phoneme could lie. This range can be interpreted as an average of the Mel-spectrograms produced by different speakers. As a result, the formants are not well separated on the frequency axis and

the phonemes have missing details. According to the author's intuition, this blurry effect occurs for most regressive model which use a simple distance based metric as loss function and could be explained by the fact that the average loss is smaller if ranges and not specific formants are predicted. This implies that an alternative metric could be helpful for regressive models.
To the best of the author's knowledge, no metric exists that is very good at measuring the quality of predicted frames. A suitable metric should take various aspects into account. It needs to be robust to shifts and dilatations on the time-axis as well as on the frequency-axis. It should also evaluate whether correct phonemes are predicted, if they are accurate and whether they are consistent with the characteristics of the speaker.
However, developing such a metric is difficult and requires a lot of expertise. Besides the development of new metrics, the following modifications could help to further reduce the existing blurry effect in the predicted Mel-spectrograms:

- Learn a suitable loss function
- Use a different network architecture

Loss Function - An alternative to the development of a new loss function could be to learn a suitable metric. For example, a Siamese network [60], [61] could be trained to compare sequences of Mel-spectrogram. Therefore, two Mel-spectrograms are fed into this additional network and the last layer outputs a similarity score. If augmented sequences are used during training, this network could learn a metric which is robust to shifts or dilatations across the time and the frequency dimensions. After training, the parameters of the Siamese network could be frozen and then used to calculate the distance between the ground truth and the predicted frame. By doing so, the L1 loss function could be replaced by a Siamese network, which was trained with a much simpler cross entropy loss (i.e. similar or not similar).
If the dataset has phoneme-level transcriptions, also a classification network could be used instead of a Siamese network. This network could be trained to predict the phonemes contained in a Mel-spectrogram. After training, the classification network could process the predicted sequences. It will only be able to classify the predicted sequences correctly if the predictions look like actual phonemes. This would allow to combine an existing distance metric with an additional classification score.

Different Architecture - The predictions could also be improved with changes to the architecture. Currently, post-processing is only used to map the latent representations to the size of the output vector. However, this could also be extended to optimize the result. For example, Wang et al. [29] used post-processing in their end-to-end speech synthesis model to convert the spectrograms from a Mel scale to a linear scale. This was done to apply the Griffin-Lim algorithm [33] to a spectrogram with linear scale. Moreover, the post-network was used to correct the predicted sequence. Since the post-processing network has access to the entire predicted sequence, it can use forward and backward information to correct the prediction error of individual frames.
An alternative to regressive models are Generative Adversarial Networks (GAN). Due to their architecture, these networks do not require the use of distance metrics to compare the prediction to the ground truth. GANs consist of two modules: The generator learns to predict plausible data, while the discriminator learns to discriminate between the prediction from the generator and the real data. The discriminator penalizes the generator for producing implausible predictions. Thus, the generator gradually improves its predictions. Since the discriminator learns to predict whether the prediction is correct, there is no need for a loss function to compare frames. Eskimenz et al. [62] have already applied GANs to genereate Mel-spectrograms and achieved realistic looking results.

A. Further Improvements

Using More Data - The proposed architecture is based on RNNs, more precisely on GRUs. A typical characteristic of RNNs is that they process the data sequentially. This means that one frame after the other is fed into the recurrent layer. The use of RNNs is feasible when using relatively small datasets like TIMIT and only few input frames n. However, in the field of text data processing, auto-regressive models have continuously used larger datasets in recent years [63]. This has mainly been enabled by using Transformer or slight variations of it such as Sparse Transformers [44] instead of RNNs. In order to develop better models for predicting speech data, it might be necessary to use more data and larger models as well. In this case, RNNs may no longer be sufficient due to computational limitations and the GRU network could be replaced by the Transformer's encoder as feature extractor. Whether a complete Transformer consisting of encoder and decoder should be used is questionable, as such sequence-to-sequence models have led to worse results as described in the appendix in chapter A-B.

Metadata - The model must not only learn to predict the correct phonemes, but also to predict them with the correct pitch and speed. These characteristics are sentence and speaker dependent. Chapter C shows that especially the speed or duration of pauses are wrong predicted and harm the performance. Adding metadata to the input vectors could help to reduce these issues. For example, it is feasible that the network can better estimate the speech rate if it knows the characteristics of the speaker or what sentence is being said.

VIII. Conclusion

The implemented model can successfully predict frames of Mel-spectrograms if previous frames are given as an input. Using Mel-spectrograms, the results suggest that frames that are closer to the given input rely more on local features, while frames that are further away rely on more global features.
The frames closer to the given data tend to be more accurately predicted. The reason is that these frames are more similar

to the last given frames due to the slow evolution of Mel-spectrograms over time. This makes their prediction simpler, as the uncertainty in the prediction process is smaller.
Furthermore, the accuracy also depends on the phoneme category and the speaker. The model predicts pause durations, closures, affricates and fricatives worse than other phoneme types. This could be due to the fact that pause durations and lip closures are highly speaker-dependent and that both affricates and fricatives contain random frequencies which are hard to predict. The accuracy is also lower when speech spoken by individuals with a lower level of education or an African-American origin is predicted. This illustrates that such systems can have issues regarding fairness and corresponding measures must be taken.
In this work, only the prediction of frames of Mel-spectrograms was investigated. The examination and application of methods for re-synthesizing Mel-spectrograms into speech was not within the scope. However, chapter II-B provides references to state-of-the-art models that could be used to re-synthesize the predictions to waveforms.
Overall, the predictions of the model are noisy. Since the model is optimized to achieve the smallest possible loss for all speakers, it predicts a cross-speaker average value for the phoneme lengths and formants. As a result, the individual formants often span multiple frequency ranges and are not separated well from each other. Therefore, the author considers the definition of appropriate metrics and loss functions as the main challenge for the task of predicting Mel-spectrograms based on regressive models. Besides defining such metrics, alternative approaches such as learning a proper metric or using alternative networks architectures could improve the result.

IX. Acknowledgement

This work was supervised by Prof. Dr. Thilo Stadelmann, Director of the Centre for Artificial Intelligence at the Zurich University of Applied Sciences. Many of his ideas have been incorporated into this project thesis and have contributed significantly to the results. Further thanks goes to Prof. Dr. Volker Dellwo, Associate Professor of Phonetics at the University of Zurich for contributing his domain knowledge and supporting the work.

References

[1] N. Singh, A. Agrawal, and P. R. Khan, "Automatic speaker recognition: Current approaches and progress in last six decades," *Global Journal of Enterprise Information System*, vol. 9, pp. 38–45, 07 2017.

[2] V. Roger, J. Farinas, and J. Pinquier, "Deep neural networks for automatic speech processing: A survey from large corpora to limited data," 2020.

[3] R. Gao and K. Grauman, "Co-separating sounds of visual objects," 2019.

[4] G. M. Bhandari, R. S. Kawitkar, and M. P. Borawake, "Audio segmentation for speech recognition using segment features," in *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India- Vol II*, S. C. Satapathy, P. S. Avadhani, S. K. Udgata, and S. Lakshminarayana, Eds. Cham: Springer International Publishing, 2014, pp. 209–217.

[5] L. Lu and H. Jiang, "Content analysis for audio classification and segmentation," *Speech and Audio Processing, IEEE Transactions on*, vol. 10, pp. 504 – 516, 11 2002.

[6] R. Karpe, "A survey :on text to speech synthesis," *International Journal for Research in Applied Science and Engineering Technology*, vol. 6, pp. 351–355, 03 2018.

[7] D. Nagalavi and M. Hanumanthappa, "N-gram word prediction language models to identify the sequence of article blocks in english e-newspapers," in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 2016, pp. 307–311.

[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[9] D. Sharma and J. Atkins, "Automatic speech recognition systems: Challenges and recent implementation trends," *International Journal of Signal and Imaging Systems Engineering*, vol. 7, pp. 220–234, 12 2014.

[10] F. Soong, A. Rosenberg, L. Rabiner, and B. Juang, "A vector quantization approach to speaker recognition," in *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 10, 1985, pp. 387–390.

[11] M. Inman, D. Danforth, S. Hangai, and K. Sato, "Speaker identification using hidden markov models," in *ICSP '98. 1998 Fourth International Conference on Signal Processing (Cat. No.98TH8344)*, 1998, pp. 609–612 vol.1.

[12] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200499903615

[13] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1695–1699.

[14] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021–1028.

[15] S. Ding, T. Chen, X. Gong, W. Zha, and Z. Wang, "Autospeech: Neural architecture search for speaker recognition," 2020.

[16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[17] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, D. E. Rumelhart and J. L. Mcclelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318–362.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[19] L. Rabiner and R. Schafer, *Theory and Applications of Digital Speech Processing*. London: Pearson, 2011.

[20] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," 2019.

[21] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," 2015.

[22] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," 2019. [Online]. Available: https://openai.com/blog/better-language-models/

[24] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[25] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," 2019.

[26] N. Kitaev, Łukasz Kaiser, and A. Levskaya, "Reformer: The efficient transformer," 2020.

[27] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," 2020.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[29] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," 2017.

[30] 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," 2018.

[31] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," 2019.

[32] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," 2015.

[33] N. Perraudin, B. Peter, and S. Peter, "A fast griffin lim algorithm," 2013. [Online]. Available: http://infoscience.epfl.ch/record/196458

[34] 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," 2016.

[35] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," 2018.

[36] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," 2020.

[37] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic Data Consortium*, 11 1992.

[38] P. Prandoni and M. Vetterli, "From lagrange to shannon...and back: Another look at sampling," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 138–144, 2009. [Online]. Available: http://infoscience.epfl.ch/record/142156

[39] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, 1965. [Online]. Available: http://www.jstor.org/stable/2003354

[40] R. B. Blackman and J. W. Tukey, *Particular Pairs of Windows: In The Measurement of Power Spectra, From the Point of View of Communications Engineering*. Dover, 1959.

[41] V. J., S. S. S., and N. E.B., "A scale for the measurement of the psychological magnitude pitch," *Journal of the Acoustical Society of America*, vol. 8, pp. 185–190, 01 1937.

[42] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," 2019.

[43] T. Gráf, "Accuracy and fluency in the speech of the advanced learner of english," 2015.

[44] R. Child, S. Gray, A. Radford, and I. Sutskever, "Generating long sequences with sparse transformers," 2019.

[45] Y.-A. Chung, Y. Wang, W.-N. Hsu, Y. Zhang, and R. Skerry-Ryan, "Semi-supervised training for improving data efficiency in end-to-end speech synthesis," 2018.

[46] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, p. 1929–1958, Jan. 2014.

[47] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[49] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.

[51] J. F. Kaiser and R. W. Schafer, "On the use of the i0-sinh window for spectrum analysis," p. 105–107, Jun. 1980.

[52] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Processing Letters*, vol. 27, p. 1485–1489, 2020. [Online]. Available: http://dx.doi.org/10.1109/LSP.2020.3016837

[53] R. Bellman and R. Kalaba, "On adaptive control processes," *IRE Transactions on Automatic Control*, vol. 4, no. 2, pp. 1–9, 1959.

[54] M. Cuturi and M. Blondel, "Soft-dtw: a differentiable loss function for time-series," 2018.

[55] J. T. Barron, "A general and adaptive robust loss function," 2019.

[56] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015.

[57] M. Stanek and M. Sigmund, "Speaker dependent changes in formants based on normalization of vowel triangle," in *2013 23rd International Conference Radioelektronika (RADIOELEKTRONIKA)*, 2013, pp. 329–333.

[58] Y. Zheng, "Acoustic modeling and feature selection for speech recognition," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2005.

[59] B. Zellner, "Pauses and the temporal structure of speech," 05 2000.

[60] R. Agrawal and S. Dixon, "Learning frame similarity using siamese networks for audio-to-score alignment," 2020.

[61] L. Nanni, A. Rigo, A. Lumini, and S. Brahnam, "Spectrogram classification using dissimilarity space," *Applied Sciences*, vol. 10, p. 4176, 06 2020.

[62] S. E. Eskimez, D. Dimitriadis, R. Gmyr, and K. Kumatani, "Gan-based data generation for speech emotion recognition," October 2020. [Online]. Available: https://www.microsoft.com/en-us/research/publication/gan-based-data-generation-for-speech-emotion-recognition/

[63] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," 2020.

[64] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[65] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014.

[66] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[67] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," 2017.

[68] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, p. 270–280, Jun. 1989. [Online]. Available: https://doi.org/10.1162/neco.1989.1.2.270

[69] A. Jongman, R. Wayland, and S. Wong, "Acoustic characteristics of english fricatives," *The Journal of the Acoustical Society of America*, vol. 108, pp. 1252–63, 10 2000.

[70] A. Ignatiev, M. C. Cooper, M. Siala, E. Hebrard, and J. Marques-Silva, "Towards formal fairness in machine learning," in *Principles and Practice of Constraint Programming*, H. Simonis, Ed. Cham: Springer International Publishing, 2020, pp. 846–867.

[71] D. Madras, T. Pitassi, and R. Zemel, "Predict responsibly: Increasing fairness by learning to defer," 2018. [Online]. Available: https://openreview.net/forum?id=SJUX_MWCZ

[72] K. Padh, D. Antognini, E. L. Glaude, B. Faltings, and C. Musat, "Addressing fairness in classification with a model-agnostic multi-objective algorithm," 2020.

[73] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "Ai fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," 2018.

[74] M. Du, F. Yang, N. Zou, and X. Hu, "Fairness in deep learning: A computational perspective," *IEEE Intelligent Systems*, pp. 1–1, 2020.

[75] K. Maughan and J. P. Near, "Towards a measure of individual fairness for deep learning," 2020.

APPENDIX A
OTHER ARCHITECTURES

Besides the proposed model based on GRUs, other architectures were also examined. However, these architectures have led to less accurate predictions. Nevertheless, they are described in this chapter in order to provide insights for any follow-up work.

A. CNN Architectures

Different architectures based on Convolutional Neural Networks (CNNs) were trained for predicting frames of speech. Among others, a U-Net [64] like architecture with a down-sampling encoder and an up-sampling decoder was trained. The CNN architectures tested generally performed worse than the architectures based on RNNs. The lower accuracy can not be directly explained by the properties of CNNs, but could also be the result of the way they were implemented. A presumption of the author is that the used receptive field was not suitable.

For example, the first predicted frames depend strongly on the last given frames. This manifests itself in the fact that the transition between the given frames and predicted frames should be smooth. The same applies also along the frequency axis. Phonemes often span a wide range of frequencies and therefore also require a large enough receptive field. Consequently, convolutional layers with large enough kernels need to be used. The activation maps only have a proper receptive field if the kernels can capture enough information from the previous layers.

Another challenge is that down-sampling reduces the feature maps. Experiments have shown that the results are worse if the feature maps become too small. A hypothesis of the author is that too much information about the temporal context as well as some part of the frequencies is lost, which results in larger prediction errors.

Overall, worse results were achieved with architectures based on CNNs. However, this could also be due to the fact that such architectures are more difficult to tune for the task of frame prediction. Thus, it cannot be concluded that CNNs in general work worse, but that their definition is more complex.

B. Seq2Seq Models

A sequence-to-sequence (seq2seq) model [65] converts a sequence of arbitrary length to another sequence of arbitrary length. These models typically consist of an encoder that maps the given sequence into a context vector and a decoder that predicts the target vector based on the context vector. Many architectures are based on RNNs and are often combined with attention mechanisms for longer sequences [66], [67]. In recent years, Transformers have become state-of-the-art for seq2seq modeling. Therefore, Transformers were also used in this work for the prediction of frames of speech.

During training, the input sequence is typically fed into the encoder and the target sequence is fed into the decoder. Thereby, the target sequence is masked and shifted by one frame, such that only the previous frames are accessible for the model. The target sequence is fed into the decoder for two reasons: First, the model learns to predict a token based on the previous tokens. The learning process is more stable if a token h_t is predicted based on the previous tokens $h_0, ..., h_{t-1}$ instead of the previous predictions $\hat{h}_0, ..., \hat{h}_{t-1}$. Second, by feeding the target sequence in the decoder, all tokens can be processed in parallel and the training time decreases. However, the target sequence is not known during inference and the prediction of the previous frame has to be re-used as an input to the decoder.

The Transformer performed very well in predicting the next frame. However, when longer sequences (i.e. more than one frame) were predicted and therefore predictions had to be reused as an input, the results were very poor. This could be due to the fact that a prediction has to be made based on more noisy inputs. In order to ensure that the model improves its prediction based on noisy inputs, teacher forcing [68] was randomly deactivated. Accordingly, either correct frames or frames from the previous prediction were fed randomly into the decoder during training. This slightly improved the results, but longer predictions were still very inaccurate. Overall, good results could only be obtained for frames directly following the given frames but not for longer sequences.

APPENDIX B
LOSS OF SHIFTED PREDICTIONS

This chapter examines the robustness of loss functions to slight shifts. Therefore, the same sequence was used as ground truth and as prediction, whereby the prediction vector was slightly shifted on the time axis or on the frequency axis. Afterwards, the loss of the shifted sequence with respect to the original was calculated. In addition, the average over the input sequence was calculated and its loss was also determined with respect to the original sequence. By comparing the loss of the shifted sequence and the loss of the mean vector, the characteristics of the loss function are investigated.

In the following the L1 loss (MAE), the L2 loss (MSE), the soft-DTW loss with MAE as the distance metric and the soft-DTW loss with MSE as the distance metric are considered. The adaptive robust loss as described in section V-A is not examined in this chapter, because this function adapts during training and an evaluation is therefore not feasible.

If the loss of a shifted vector is larger than the loss of the mean vector, then the loss function is considered not robust enough to the corresponding shift. This is due to the fact that the model achieves a smaller loss when an average value is predicted instead of the correct sequence that is shifted.

Figure 7 shows the loss value of different loss functions for the ground truth and the same vector shifted by $1, 2, 4, 6, 10$ and 20 frames on the time axis. The grey background indicates the loss of the mean vector compared to the ground truth. It can be observed that the L1 loss is more robust against shifts on the time axis than the L2 loss. For example, if the vector is shifted by 4 frames on the time axis (green line), the L1 loss of the shifted vector is on average smaller than the L1 loss of the mean vector. For the L2 loss, on the other hand, the average

loss of a vector shifted by 4 frames is larger than the loss of the mean vector. The two soft-DTW losses show similar behavior. When the MAE is used as a distance metric, the shifted predictions achieve on average better results in relation to the mean vector than when the MSE is used as a distance metric.
The plot also shows that a lower limit of number of frames to predict exists when using soft-DTW as loss function. If only a few frames are predicted (i.e. left range on the x-axis), then the loss of slightly shifted predictions is often larger than the mean vector. This indicates that soft-DTW should only be used if longer sequences are predicted.

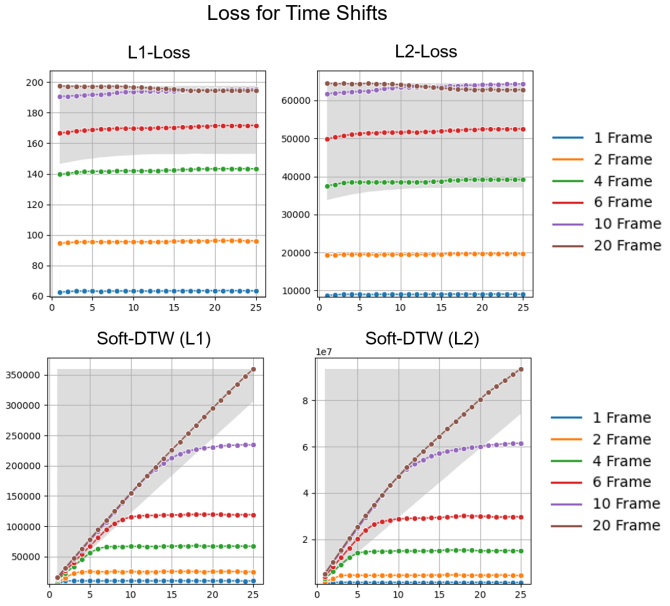

Fig. 7. The average loss of sequences which were shifted by different numbers of frames on the time axis. The y-axis shows the loss, the x-axis the sequence length of the prediction in frames. The gray background marks the area with a higher loss than a mean vector.

The same analysis was done for shifts on the frequency axis. Figure 8 shows the loss between the ground truth and the same vector shifted by $100, 200, 300, 400$ and 500Hz. It can be observed that slight shifts on the frequency axis are in general less affected by the problem that they have a larger error than the mean vector.

Based on these findings, the L1 loss is preferred, as it is more robust than the L2 loss for shifts on the time axis. Unlike the soft DTW losses, the L1 loss is also suitable for the prediction of shorter sequences.

Appendix C
Error per Phoneme

The model learned to classify some sentences better than others. One reason for this effect is that the network is better at predicting certain phonemes than others. The TIMIT corpus contains time-aligned phonetic transcriptions which were used to evaluate this behaviour. In the TIMIT dataset, the phonemes are categorised as stops, affricates, fricatives, nasals,

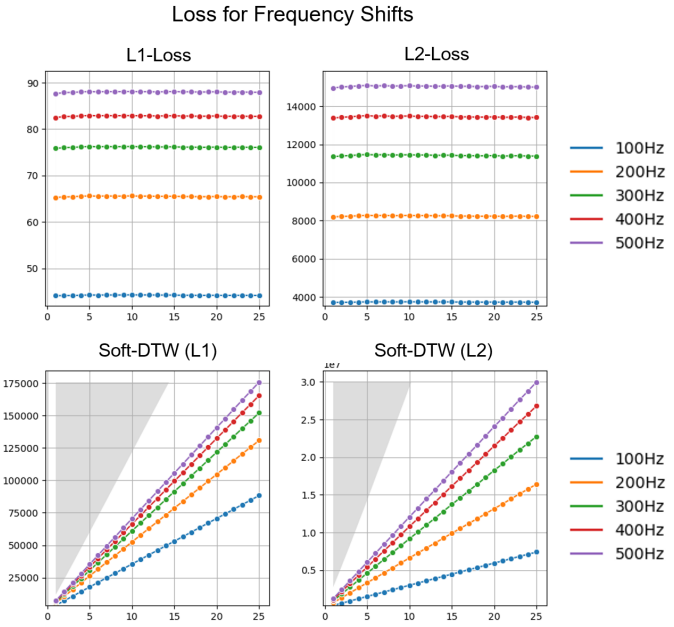

Fig. 8. The average loss of sequences which were shifted by different frequencies on the frequency axis. The y-axis shows the frequency, the x-axis the sequence length of the prediction in frames. The gray background marks the area with a higher loss than an average noise vector.

semivowel, glides and vowels. In addition, the lip closure intervals of stops and affricates are transcribed individually. The additional category "others" contains the transcriptions for the start and end of sentences as well as pauses.

Figure 9 shows the MAE per phoneme. Additionally, the colors indicate which phonemes belong to which category. It can be seen that the model has the highest error for closures of the lips near stops and near affricates. This is due to the fact that it is difficult to estimate transitions near a lip closure as Ref. [58] has shown.
The same applies to the detection of stops in general. Especially pauses (label "pau") between individual phonemes are poorly predicted. The reason is that pause durations are context and speaker dependent [59] and thus has a high variance.
Besides closures of lips and pauses, affricates and fricatives have a larger MAE on average than the other phonemes. Affricates are sounds made up of a stop, immediately followed by a fricative. A fricative, on the other hand, includes by definition an occlusion or obstruction in the vocal tract great enough to produce noise (frication). This process is not only speaker dependent, but the generated air stream of fricatives creates a mix of random frequencies, lasting only a short time [69]. Therefore, these types of phonemes are more difficult to predict because they contain a random component.

Appendix D
Error per Speaker

Besides different errors per sentence, also a variance per speaker is observable. The TIMIT dataset provides additional information on the 630 speakers. Four characteristics that are

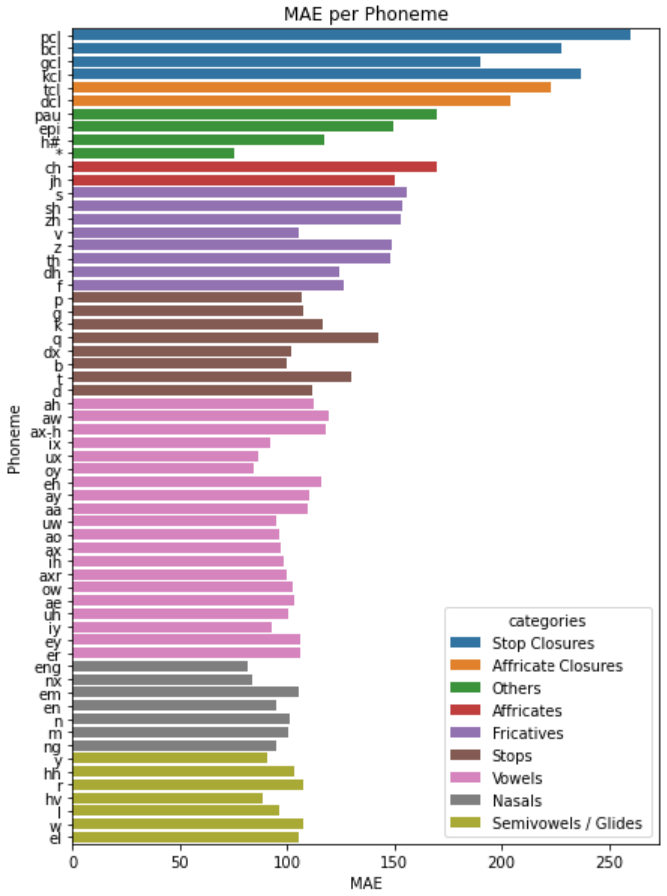

Fig. 9. The mean absolute error (MAE) per phoneme according to the TIMIT phoneme definition. The phonemes are grouped into categories "stop closures", "affricate closures", "others", "affricates", "fricatives", "stops", "vowels", "nasals" and "semivowels and glides".

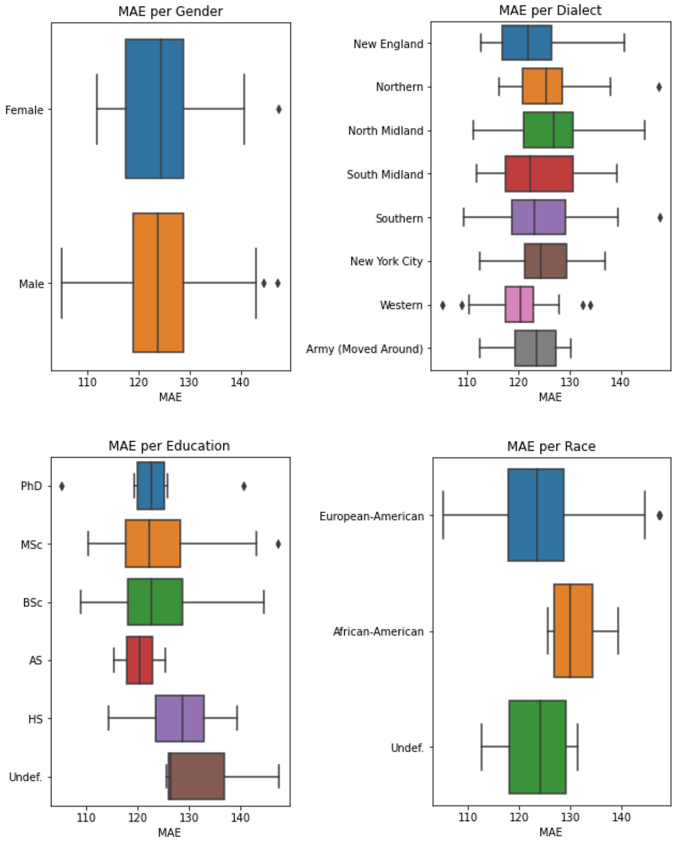

Fig. 10. The Mean Absolute Error (MAE) for various speaker characteristics. Top left shows the MAE differentiated by gender, top right shows the MAE by dialect region, bottom left the MAE by the speaker's education, and bottom right shows the MAE by race.

provided per speaker are gender, dialect, education, and race. A important requirement of artificial intelligence (AI) systems in general is fairness [70], [71]. A system should not discriminate against groups of individuals based on attributes such as race, gender or education [72].

The evaluations as shown in figure 10 suggest that the model has issues regarding fairness. The model works about equally well for men and women, although there are twice as many recordings from men as from women in the dataset. This indicates good generalization with respect to gender. Likewise, all dialect regions are recognized about equally well.
However, some issues arise regarding education and race. For example, speech of people with lower levels of education (i.e. High School or undefined) is predicted worse than speech of people with higher levels of education (i.e. Associate degree, Bachelor's degree, Master's degree or PhD). With respect to race, speech frames spoken by African Americans are predicted worse than if they are spoken by European Americans.
Solving these problems is outside the scope of this work. Various approaches on how to tackle such issues can be found in the literature [73]–[75]. However, the purpose of the evaluation in this section is to draw attention to this problem. Such systems should be evaluated for fairness before they are deployed and used in production.