Project Thesis 1: Towards Predicting Speech with Sequence Learners*

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

***Abstract*—Research in the area of speech data analysis with deep learning is mainly focused on speech recognition, speech segmentation, speaker recognition and speech synthesis. The prediction of frames of speech is a less studied area so far. In this thesis, it is shown that a neural network based on Gated Recurrent Units (GRUs) can be used to predict frames of speech up to one word in length. More precisely, a model is proposed and trained on the TIMIT data-set to predict for a small number of given frames of a Mel-spectrogram the subsequent frames. The predictions are examined in detail and strengths as well as weaknesses of the system are highlighted. Results are presented on how many frames should be used as input and how high the accuracy of the prediction per frame and per phoneme is. As this is one of the first works in this area, especially the lessons learnt are pointed-out and possible improvements for future works are proposed. One of the main challenges that such systems face is the definition of a suitable loss function. A well-suited loss function should have high robustness when comparing predicted frames with target frames. Furthermore, it is shown that the system exhibits different characteristics depending on the used distance metric as loss function.**

***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 are scientifically well-studied areas, for example speaker recognition [1], speech recognition [2], audio separation [3], audio segmentation [4], audio classification [5] or text to speech conversion [6]. What has been less studied so far is the prediction of speech. In the field of Natural Language Processing (NLP), the prediction of subsequent words has been researched for quite some time and is used, for example, for word correction and spell check systems [7] as well as to calculate vector representations of words [8]. In the field of audio data, however, this is not the case.

Processing audio signals is complex and characterised by various challenges. For example, such systems should 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 today to process audio data.

In this project thesis, the prediction of frames of a Mel-spectrogram [19] is investigated using deep learning methods. Thereby, the prediction is done without conversion to text (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, it is an interesting problem itself and there is a practical application of it, for example to optimize audio interfaces by completing truncated signals. Second, such systems can be used to extract features from Mel-Spectrograms which can then be used for other tasks such as speaker classification II-A. Feature extraction systems are usually optimized for the following downstream task. In this work, the focus is on predicting frames of speech as accurate as possible and not to generate features for other tasks.

II. Related Work

In this section, relevant work is presented. Particularly relevant are considered auto-regressive models for generating speech representations as well as speech synthesis models.

* This is the documentation of the first project. It 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 project thesis was supervised by Prof. Dr. Thilo Stadelmann. The code is publicly available on Github https://github.com/sagerpascal/temporal-speech-context.

A. Auto-Regressive Models for Speech Representation Learning

In the field of natural language processing (NLP), auto-regressive models are often used for unsupervised pre-training [20]. 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 [21]–[23], Transformer-XL [24] or Reformer [25].
Recently, this concept has been applied to audio data as well [26], [27]. Thereby mainly Mel-spectrograms are used and based on given frames a subsequent frame is predicted. This pre-training takes place on very large data sets like LibriSpeech [28]. The goal of the pre-training is to learn speech representations that can be used for different downstream tasks across different data-sets. 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, these models are not optimized to predict frames as accurately as possible, but 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 given frames 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], first 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 relevant because even tough they have a different input (i.e. text), they also generate Mel-spectrograms. Thus, similar principles can be applied for the generation process.

Additionally, speech synthesis models are also relevant when predicted Mel-specotrgrams are to be re-synthesized into waveforms. Although the re-synthesizing process is out of scope of this project thesis, relevant work that can be used for this purpose is referenced in the following.
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 state that the model is not feasible for speech prediction due to the lack of long range coherence, because 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] text corpus are used. This data-set includes recordings of 630 speakers of eight major dialects of American English reading ten out of 2342 different phonetically rich sentences. The TIMIT data-set 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 this information, the speech signal is transformed into the frequency domain.
The transformation from the time domain to the frequency domain is done by the Fast Fourier Transformation (FFT) [39]. However, this transformation requires the signal to be static. This is achieved by splitting the quasi-stationary speech signal into frames of 25ms length using the Hann window function 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 frequencies for a single frame (i.e. one sequence extracted with the Hann window). Several spectrums are calculated by shifting the Hann window by 50% 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 then 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. [40] proposed the Mel scale which scales the signal such that equal distances in pitch sound equally distant to the listener. This scale is also helpful to interpret the spectrogram and and was therefore in this work applied on the spectrogram. Spectrograms with this scale are called Mel-spectrograms and show the amplitude per frequency over time with a Mel scale.
Figure 1 depicts the entire concept of the application in an

overview. 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.

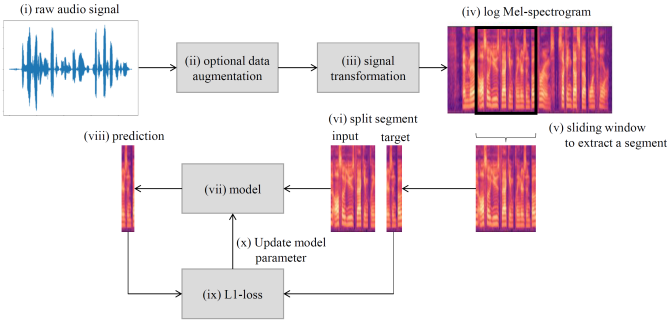

Fig. 1. The concept of this work to predict frames: First, the raw audio signal is loaded (i) and optionally data augmentation (ii) is applied. Then the audio signal is transformed (iii) to a Mel-spectrogram (iv). A sliding window (v) is used to extract a segment, which is then split into input and target (vi). A model (vii) is trained to predict the target sequence (viii) based on the input sequence. The model parameters are then updated (x) by comparing the target sequence with the prediction using the Mean Absolute Error 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 s. The extracted segment is then split into two sub-segments (vi). One segment contains n frames which are fed as input into the model. The other segment consisting of k frames is used as target of the network. The network is then trained to predict the unseen segment consisting of k frames based on the given segment with n frames. The split of the extracted segment into sub-segments can be done in three different ways:

- Predict the k subsequent frames of n given frames
- Predict the k previous frames of n given frames
- Predict the k frames between $\lceil \frac{n}{2} \rceil$ previous frames and $n - \lceil \frac{n}{2} \rceil$ subsequent frames

Since most use-cases such as reconstructing truncated signals are based on the prediction of subsequent frames, the first variant is used. However, it is quite likely that the third variant could work better, since there is information available how the predicted frames must look like at the beginning and at the end in order to have a smooth transition to the given frames. For the other two methods, this information is only available either for the beginning or for the end of the prediction.
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 [41].
In this work, the window for extracting segments has been shifted by one frame. This implies that target vectors y_{t_1} are reused in a next training sample as input vectors x_{t_2}. However, this is not considered to be a problem. Also in NLP are similar training methods applied, for example the training methods of word2vec models [8] such as c-bow or skip-grams 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 result. 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 [42] measured that a native English speaker says 196 words per minute. With this assumption it can be calculated that one word corresponds to 306ms as shown in equation 1.

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

By a given sample rate of $f_s = 16\text{kHz}$, 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 about one word of speech.

$$k_{1\text{w}} = t_{1\text{w}} \cdot f_s / \frac{h_l}{h_l/h_s} = 0.306\text{s} \cdot 16000\text{s}^{-1} / \frac{400}{400/200} \approx 24.5 \tag{2}$$

The second parameter n which determines how many frames should be fed into the model was treated as a hyper-parameter. If more frames are fed into the model (i.e. n is larger), then 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 huge impact on the performance of the entire model. Another disadvantage of longer input sequences is that it reduces the number of training samples. When using shorter sequences, more segments can be extracted from the Mel-spectrograms. Longer sequences (i.e. n is larger), on the other hand, result in fewer segments and consequently fewer training samples.

Auto-regressive language 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 [23], [24], [43] to estimate the probability distribution over the tokens. However, for speech data, each token t_k corresponds to a frame rather than a word. Since the set of target tokens for speech data is not finite, the Softmax layer is replaced with a regression layer. This means that the model for n given frames directly predicts the k subsequent target frames and does not calculate a probability for all existing frames. The model is then optimized by minimizing the L1 loss between the prediction and the ground

truth, as it is done in many speech synthesis models [29], [44]. 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 with the length n is used as input for the model and the second vector of length k as 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. [26] 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 [45] and layer normalization [46]. This pre-net is used to extract features from the Mel-spectrogram. Thereby, the first fully connected layer increases the number of channels from 80 to 512. All the other layers keep the dimensionality of 512 and thus the pre-net calculates latent representations with a dimensionality of $[512 \times n]$.

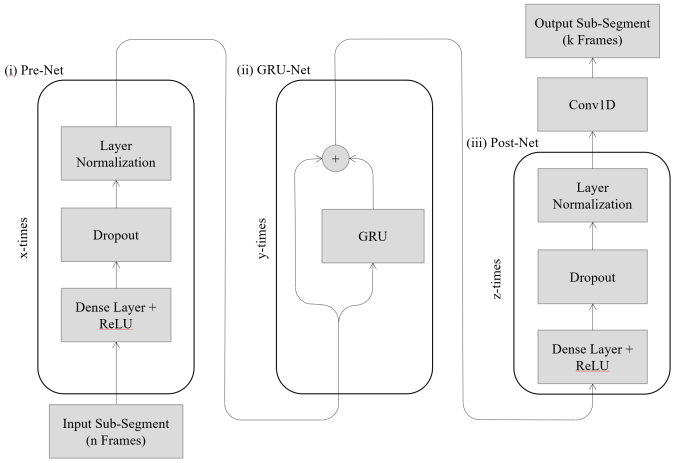

Fig. 2. The input sequence is first fed into a pre-net (i), which consists of 5 identical blocks, each containing a fully connected layer, followed by a dropout layer and layer normalization. The extracted latent representations are then fed into 4 GRUs (ii) with residual connections around it. Finally, 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 then processes the generated latent representations. It consists of 4 Gated Recurrent Units with residual connections [47] 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 [48]. In addition to residual connections, gradient-clipping with a clip coefficient of $c_{coeff} = 1$ was used to further mitigate these issues.

Afterwards, 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 number of channels from 512 to 80.

The model was trained with the Adam optimizer [49]. 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 CNN and sequence-to-sequence modeling were also evaluated. However, these architectures have not worked well 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 the Mel-spectrogram. It is important that the augmentation does not disrupt the raw signal too much, otherwise the phonemes would not be clearly identifiable in the Mel-spectrogram anymore 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 for band-limited sinc interpolation. It was used with a probability of $p_{\text{resample}} = 0.75$ and resampled the original signal by a factor in the range $[0.7, ..., 1.3]$.

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 in the range $a = [0.8, ..., 1.2]$. Since the TIMIT corpus is relatively small, data augmentation is considered necessary to achieve good results on the test-set.

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 such as speech rate and pitch based on speaker-dependent features. 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 data-sets with different speakers.

V. Preliminary Experiments

A. Loss Function

The used loss function has a huge influence on the characteristics of the prediction. In the field of speech synthesis, the L1 loss is often used to optimize the model [29], [44]. The

same applies for models which reduce noise in frames of Mel-spectrograms [50]. 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

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 [51]. For example, the model could get a high loss if the prediction is good, but not well aligned on the time axis. To address this problem, soft dynamic time warping (soft-DTW) [52] was used for the L1 loss as well as for the L2 loss. Soft-DTW is based upon dynamic time warping (DTW) [51]. 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 [53] 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 [54]. 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. As figure 3 shows, all of these loss functions produced very similar results. 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 emphasized). The L1 loss and the two Soft-DTW versions, on the other hand, emphasize these upper frequencies stronger. The fact that the soft-DTW version of the L2 loss emphasizes this 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

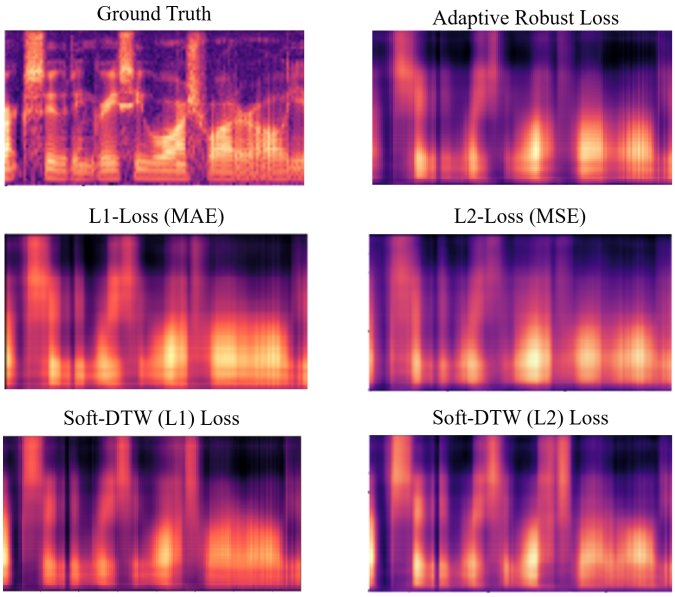

Fig. 3. The ground truth of a randomly chosen sentence and five corresponding predictions from models trained with a different loss function. In order to compare the whole sentence and not only a short sequence, several predictions were concatenated.

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, more important than the visual evaluation of Mel-spectrograms is the acoustic perception. 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. Consequently, this loss function was used for the project thesis but should be reconsidered in future work.
The loss functions are further analyzed in the appendix in chapter B. By comparing time-shifted predictions with noise, 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. These findings support the choice of the MAE as loss function.

VI. Results

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

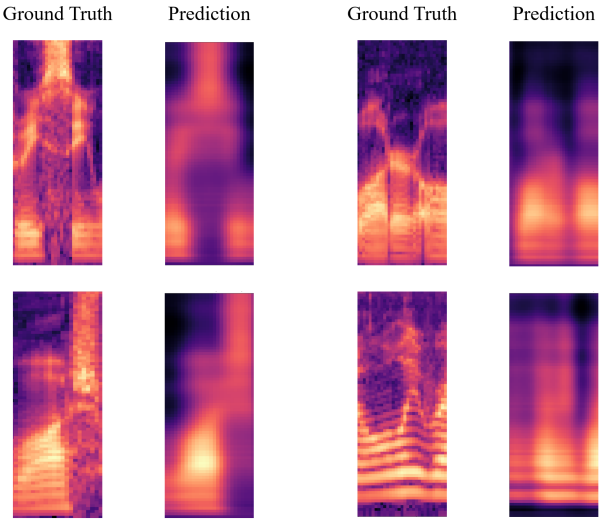

Fig. 4. Four random examples of predicted frames. The ground truth is shown on the left and the prediction of the model on the right.

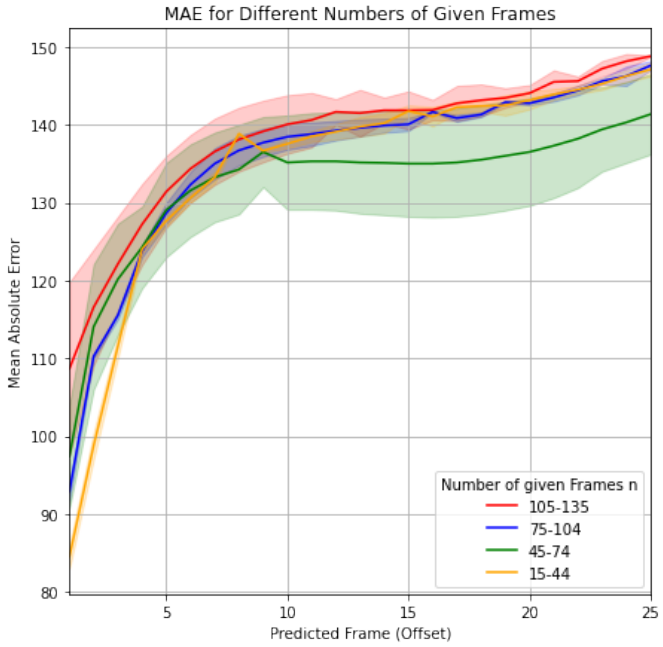

Fig. 5. The Mean Absolute Error (MAE) per predicted frame for various numbers of given frames. The y-axis shows the MAE, the x-axis show the predicted frame (e.g. 3 means the error for the 3rd frame) and the lines represent the error for different numbers of given frames.

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 from the predicted phonemes is often only the outline 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 [55]. For different speakers, this leads to different positions of the formants on the x-axis (time) and the y-axis (frequency) of the Mel-spectrogram, even if the same sentence or the same word is said. This makes the prediction task for the model difficult. By using 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 prediction 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 given 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 then the MAE of each of the 25 predicted frames was examined. Figure 5 shows the result.

The prediction accuracy of the first three frames was higher when only a few frames (n in the range 15-44 frames) were fed into the model. This means that just 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 less time-steps away from the given data are based on more local information (i.e. more dependent on the immediately preceding frames).

Predicted frames that are further away in time require more given data. For example, frames that follow 10-25 time-steps after the given data achieved better results when between 45 and 74 frames were given. This suggests that the predicted frames which lay 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 best results were obtained when about 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 given.

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

Overall, the frames that are less time-steps away from the given data have a lower MAE than the frames that are more time-steps away. This is because the frames of a Mel-spectrogram evolve slowly over time. Therefore, 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. Then the sliding window was shifted forward by one frame on the time axis and the process was repeated. At the end, all kept frames were stacked and

thus the resulting Mel-spectrogram shows the prediction with a specific offset from the given data.

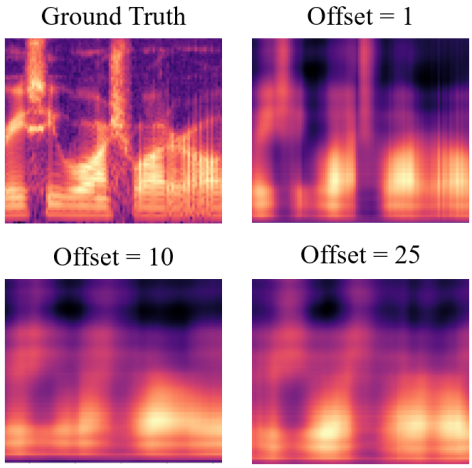

Fig. 6. The plot in the upper left corner shows the Mel-spectrogram of a randomly chosen sentence. The other plots show predictions with a specific offset. For this purpose, only the frame of a prediction that is a specific number of time-steps (offset) away from the given data was kept. Finally, several predictions were made and all kept frames were concatenated.

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 more blurry. Consequently, the accuracy per predicted frame can not only be measured in numbers using the mean absolute error, but is indeed visible on the plotted Mel-spectrograms.

B. Error per Phoneme and Error per Speaker

The TIMIT data set provides various additional information about the audio sequences. For example, time-aligned phonetic transcriptions and additional information about the speakers are given. The phonetic transcriptions were used 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 are incorrectly predicted. This could be due to the facts that it is difficult in general to estimate transitions near a closure [56] and the pause duration is highly speaker dependent [57]. The model also has more difficulty predicting affricatives and fricatives than other phoneme types. A reason for this could be that affricatives and fricatives are generally difficult to determine, as their frequencies contain a random component by definition.
In chapter D of the appendix is the prediction error per speaker examined. 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 worse predicted than frames spoken by individuals with a higher level of education or a 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/temporal-speech-context/results.html#predictions.
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 more time-steps a predicted frame is away 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 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. If this prediction is reused 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 input. Therefore, the input was 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. In chapter VI-A was shown that the first few frames have a much smaller prediction error than the frames which are more time-steps away from the given data. Therefore, less frames were predicted and used as 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 inputs 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 NLP for the prediction of longer sequences [8]. The more time-steps away from the given seed a prediction is, the higher becomes the uncertainty. Consequently, the frames that are less time-steps away from the seed are predicted more accurately. These frames are assigned a higher weight so that they have more influence on the overall loss.
All these measures have contributed to improve the model. The model is in some cases able to predict longer sequences if a seed from the training data-set is used. However, with seeds from the test-set this only works 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 clearly separated on the frequency axis and the phonemes are not separated on the time axis. According to the author's intuition, this blurry effect occurs for most regressive model which use a simple distance based metric as loss function, because the average loss is smaller if ranges and not specific formants are predicted. This implies that for regressive models an alternative metric could be helpful.
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. For example, it should be robust to shifts and dilatations on the x-axis (time axis) as well as on the y-axis (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 also help to 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 [58], [59] 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 data-set 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 looks like actual phonemes. This would allow an existing distance metric to be combined with the classification error.

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 mainly done to apply the Griffin-Lim algorithm to a spectrogram with linear scale. However, another motivation was to use the post-network 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 for 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 (i) learns to predict plausible data, while the discriminator (ii) 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 learns to produce better and better 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. [60] 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. With a relatively small data-set like TIMIT and only few given frames n, not that much computational resources are needed. However, in the field of NLP, auto-regressive models have been using more and more data in recent years [61]. This has mainly been enabled by using Transformer or slight variations of it such as Sparse Transformers [43] 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. In chapter C is shown that especially the speed or wrong pause duration can harm the performance. Adding metadata to the input vectors could help to reduce these

problems. For example, it is feasible that the network can better estimate the speech rate if it knows who the speaker is or what sentence being said is.

VIII. Conclusion

In this thesis, it was shown that frames of speech can be predicted if previous frames are given. The proposed model processes frames of Mel-spectrograms. However, besides Mel-spectrograms other representations such as Mel Frequency Cepstral Coefficients (MFCCs) [12] could also work.

By using Mel-spectrograms, charactersitics of such systems have been investigated. It was shown that frames that are less time-steps further from the given input rely more on local features, while frames that are more time-steps 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.

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, the chapter II-B provides references to state-of-the-art models that could be used to re-synthesize the predictions to speech.

Overall, the predictions of the model are noisy. Especially the phoneme length and the precise formant frequencies are predicted bad, because they are highly speaker dependent. 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 main challenge for the task of predicting Mel-spectrograms based on regressive models the definition of appropriate metrics and loss functions. Besides defining such metrics, alternative approaches that could improve the result are proposed in chapter VII.

IX. Acknowledgement

This work was supervised by Prof. Dr. Thilo Stadelmann, Head of 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] A. M. Dai and Q. V. Le, "Semi-supervised sequence learning," 2015.

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

[22] 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/

[23] 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.

[24] 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.

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

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

[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] 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.

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

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

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

[44] 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.

[45] 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.

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

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

[48] 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.

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

[50] 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

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

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

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

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

[55] 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.

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

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

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

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

[60] 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/

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

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

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

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

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

[66] 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

[67] 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.

[68] 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.

[69] 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

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

[71] 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.

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

[73] 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 worse results. 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 [62] like architecture with a down-sampling encoder and a up-sampling decoder were trained. The CNN architectures tested generally performed worse than the architectures based on RNNs. However, this must not be due to the properties of the CNNs, but could be the result of the way they were implemented. A presumption of the author is that the used receptive field was not suitable. In general, CNNs with appropriate characteristics seem to be harder to define than fully connected networks or RNNs.

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. This results in the need to use convolutional layers with large enough kernels. Only if the kernels can capture enough context the activation maps have a proper receptive field.

Another challenge is that with down-sampling the feature maps become smaller. Experiments have shown that the results become bad if the feature maps becomes too small. A hypothesis of the author is that in this case 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 proper 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 [63] 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 for longer sequences often combined with attention mechanism [64], [65]. 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, typically the input sequence is fed into the encoder and the target sequence is fed into the decoder. Thereby, the target sequence is masked and shifted by one frame, so that only the previous frames are accessible for the model and not just the identity function must be learned. 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 correct previous tokens $h_0, ..., h_{t-1}$ and not on the previous predictions $\hat{h_0}, ..., \hat{h_{t-1}}$. Secondly, by feeding the target sequence in the decoder, all tokens can be processed in parallel and the training is much faster. However, the target sequence is not known during inference and the prediction of the previous frame has to be re-used as 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 a prediction had to be reused as 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 address this problem, teacher forcing [66] was randomly deactivated. Thereby, randomly either correct frames or frames from the previous prediction were fed into the decoder during training. The aim of this process was that the model learns to predict subsequent frames based on previous predictions which could be noisy. This slightly improved the results, but longer predictions were still very inaccurate. Overall, good results could only be obtained for frames directly following on 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. It is considered a problem if a slight shifted prediction has a larger loss than, for example, an average noise. 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 was calculated between the original sequence and the shifted sequence. In addition, the prediction was compared to a noise vector. This noise vector was calculated by taking the average value of the frames. By doing so, the characteristics of the loss function is 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 were 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.

The average noise vector used for comparison corresponds to the average value of the ground truth vector. If the loss of a shifted vector is larger than the loss of this noise vector, then the corresponding shift is evaluated as too large, because the network could achieve a smaller loss with such an average value as prediction than with a correct prediction 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 average noise 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), it has on average a smaller L1 loss than the noise vector. For the L2 loss, on the other hand, the average loss of a vector shifted by 4 frames is larger than the noise vector. The two soft-DTW losses also exhibit similar behavior. When the MAE is used as distance metric, the shifted predictions achieve on average better results in relation to the noise vector than when the MSE is used as distance metric.
The plot also shows that there exists a lower limit of number of frames to predict 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 average vector. This indicates that soft-DTW only should be used if longer sequences are predicted.

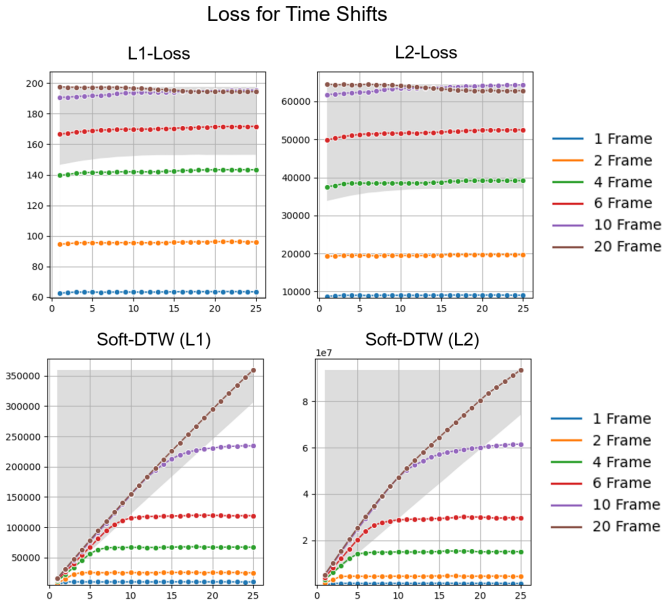

Fig. 7. The average loss of sequences which where shifted by different numbers of frames on the time axis. The y-axis shows the loss, the x-axis the sequence length of the prediction in frames. The gray background marks the area with a higher loss than an average noise vector.

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 in general slight shifts on the frequency axis are less affected by the problem that they have a larger error than an average noise.

Based on these findings, the L1 loss is preferred, as it is more robust than L2 loss for shifts on the time axis. Moreover, the L1 loss is unlike the two soft-DTW losses 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

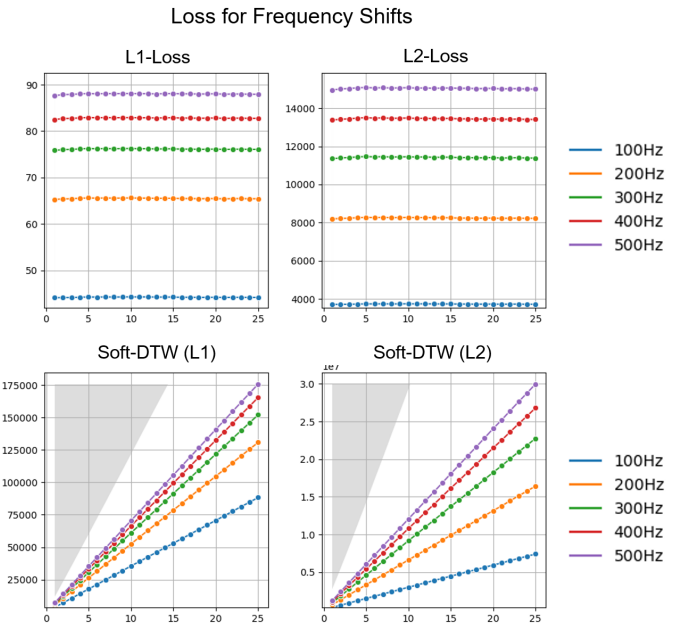

Fig. 8. The average loss of sequences which where shifted by different frequencies on the frequency axis. The y-axis shows the frequency, the x-axis the sequence length of the prediction in frames. The gray background marks the area with a higher loss than an average noise vector.

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 data-set, the phonemes are categorised as stops, affricatives, fricatives, nasals, semivowel, glides and vowels. In addition, the closure intervals of stops and affricatives are transcribed individually. The additional category “others” contains the transcriptions for the start and end of sentences as well as pauses.

The 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 the closures of stops and closures of affricatives. This is due to the fact that it is difficult to estimate transitions near a closure as Ref. [56] has shown. For example, the timing when the next phoneme begins is often predicted incorrectly.
The same applies to the detection of stops in general. Especially pauses (label “pau”) between individual phonemes are poorly predicted. This is due to the fact that the pause duration is context and speaker dependent [57] and thus has a high variance.
Besides closures and pauses, affricatives and fricatives also have on average a larger MAE than the other phonemes. Affricatives 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 strongly speaker dependent, but the generated air stream of fricatives creates a mix of random frequencies, lasting only a short time [67]. Therefore, these types of phonemes are more difficult to predict and results in a higher error compared to

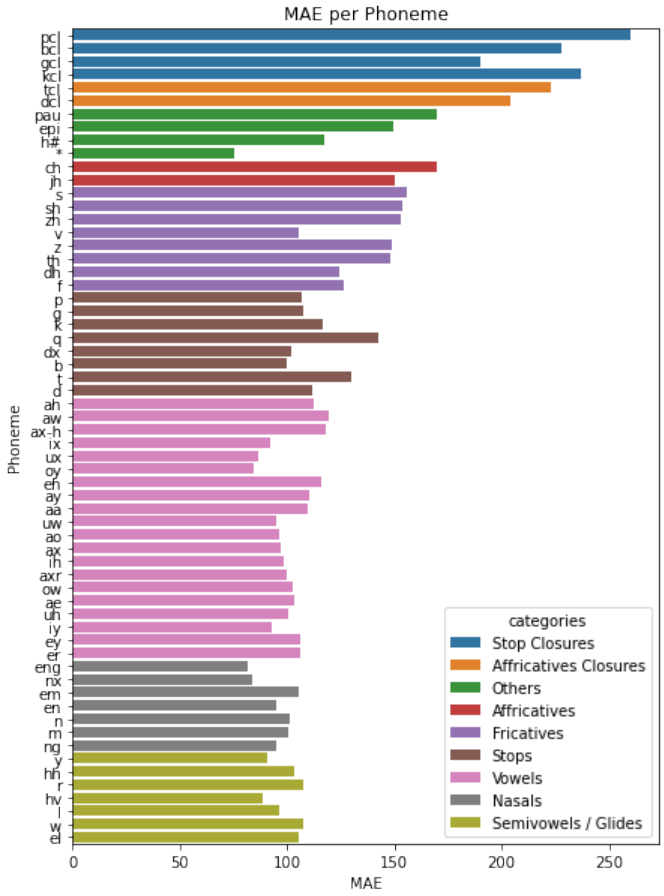

Fig. 9. The mean absolute error (MAE) per phoneme according to the TIMIT phoneme definition. The phonemes are grouped into categories "stop closures", "affricative closures", "others", "affricatives", "fricatives", "stops", "vowels", "nasals" and "semivowels and glides" (from top to bottom).

other types of phonemes.

Appendix D
Error per Speaker

Besides different errors per sentence, also a variance per speaker is observable. The TIMIT data-set provides additional information on the 630 speakers. Four characteristics that are provided per speaker are gender, dialect, education, and race. A very important property of artificial intelligence (AI) systems in general is fairness [68], [69]. This means that a system does not discriminate against groups of individuals based on attributes such as race, gender or education [70].

The model presented here is not intended for productive use. The evaluations as shown in figure 10 suggest that the model is also not suitable for this purpose, due to 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 data-set. 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.

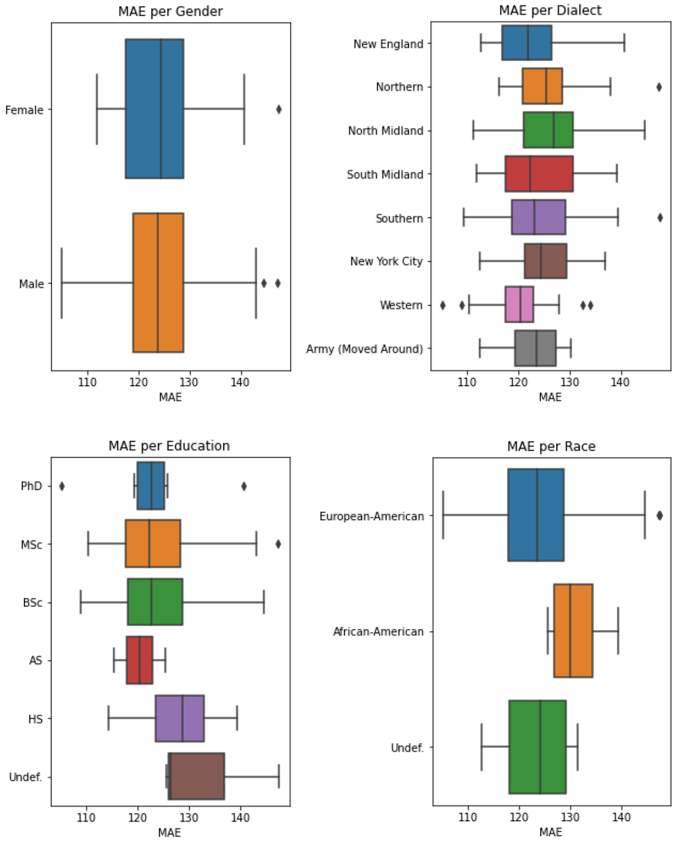

Fig. 10. The Mean Absolute Error (MAE) for various speaker characteristics. Top left shows the MAE differentiated by gender, top right shows the MAE by dialect region, bottom left the MAE by the speaker's education, and bottom right shows the MAE by race.

High School or undefined) is predicted worse than speech of people with higher level 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. In the literature exist various approaches on how to tackle such issues [71]–[73]. 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.