Estimating articulatory movements in speech production with transformer networks

Sathvik Udupa[1], Anwesha Roy[1], Abhayjeet Singh[1], Aravind Illa[2]*, Prasanta Kumar Ghosh[1]

[1]Electrical Engineering, Indian Institute of Science (IISc), Bangalore-560012, India
[2]Amazon Alexa, Bangalore, India

Abstract

We estimate articulatory movements in speech production from different modalities - acoustics and phonemes. Acoustic-to-articulatory inversion (AAI) is a sequence-to-sequence task. On the other hand, phoneme to articulatory (PTA) motion estimation faces a key challenge in reliably aligning the text and the articulatory movements. To address this challenge, we explore the use of a transformer architecture - FastSpeech, with explicit duration modelling to learn hard alignments between the phonemes and articulatory movements. We also train a transformer model on AAI. We use correlation coefficient (CC) and root mean squared error (rMSE) to assess the estimation performance in comparison to existing methods on both tasks. We observe 154%, 11.8% & 4.8% relative improvement in CC with subject-dependent, pooled and fine-tuning strategies, respectively, for PTA estimation. Additionally, on the AAI task, we obtain 1.5%, 3% and 3.1% relative gain in CC on the same setups compared to the state-of-the-art baseline. We further present the computational benefits of having transformer architecture as representation blocks.

Index Terms: electromagnetic articulograph, acoustic to articulatory inversion, phoneme to articulatory estimation, transformer network

1. Introduction

Knowledge about the articulatory position along with the acoustics is useful in applications like automatic speech recognition [1, 2], language learning [3, 4] and speech synthesis [5, 6]. In practice, in the absence of direct articulatory movements, they are typically estimated from different modalities like acoustic features (Mel-Frequency Cepstral Coefficients (MFCC)) and text (phoneme sequence). A rich literature exists related to the estimation of articulatory movements from acoustic features of speech, known as acoustic-to-articulatory inversion (AAI). Various approaches have been proposed, including Gaussian Mixture Model (GMM) [7], Hidden Markov Model (HMM) [8], and neural network [9, 10]. The state-of-art performance is achieved by the bidirectional long short-term memory (BiLSTM) networks [11, 12].

On the other hand, there have been few attempts on phoneme to articulatory (PTA) mapping even though it has many potential applications like in language tutoring systems for learning correct pronunciation or for analyzing pronunciation defects [13]. These attempts typically used the techniques from the speech synthesis paradigm like HMM [13] and BiL-STM [14]. Works in [13, 14] reported a considerable drop in phonological feature (PTA model) performance compared to

that using acoustic features (AAI model). One reason for the decline in performance using phonemes compared to acoustic features could be due to the limitations of the duration modelling with HMM [13, 14]. Hence, there is a need for effective duration modelling to obtain the articulatory frame length for a given phoneme sequence. In [15], the Tacotron [16] speech synthesis model is deployed for the PTA task. Using the duration modelling in Tacotron, a soft alignment between the phonemes and the articulatory movement sequence is obtained. This is done by learning the attention weights, which implicitly model the time alignment between an encoder (phoneme representation) and decoder (articulatory representation) hidden states. However, due to the implicit duration modelling in Tacotron, it leads to poor convergence with a limited amount of training data [15]. Further, sequential computations involved in the Tacotron network result in longer training and inference time.

To overcome these limitations, this work explores the transformer networks for estimating articulatory movements. The transformer [17] is a state-of-the-art sequence-to-sequence transduction model with an encoder-decoder structure. The architecture was first introduced on neural machine translation tasks, and recently, various transformer architectures [18, 19, 20] have performed well in speech synthesis. We use a non-autoregressive transformer that exploits the entire sequence simultaneously. This allows for significantly more parallelization within training examples, which is important for longer sequences. We deploy the architecture presented in FastSpeech [19] for the PTA task. This allows for explicit duration modelling allowing hard alignment between the phonemes and the articulatory movement frames.

In addition to the PTA task, we also use a transformer model for the AAI task. To the best of our knowledge, this is the first work to utilize transformer networks for articulatory estimation. We perform extensive experiments with different positional encoding schemes and evaluate the results on different experimental setups. We compare the proposed approach with baseline models, and experimental results demonstrate that transformers achieve 154%, 11.8% and 4.8% relative improvement in terms of correlation coefficient (CC) on subject-dependent, pooled and fine-tuned setups, on the PTA task, while the relative gains in CC are, respectively, 1.5%, 3% and 3.1% on the AAI task. The code for the experiments can be found at https://github.com/bloodraven66/aai_pta_with_transformers.

2. Dataset

A set of 460 phonetically balanced English sentences are considered from the MOCHA-TIMIT corpus as the stimuli for data collection from 10 subjects comprising of 6 male (M1, M2, M3, M4, M5, M6) and 4 female (F1, F2, F3, F4) subjects with ages in the range of 20-28 years. All the subjects are native Indians

---

who are proficient in English. We simultaneously recorded audio signals using a microphone [21] and articulatory movement data using Electromagnetic Articulograph (EMA) AG501 [22]. EMA AG501 has 24 channels to measure the horizontal, vertical, and lateral displacements as well as angular orientations of a maximum of 24 sensors. The sensors were placed according to the guidelines provided in [17] and the articulatory movement was recorded at a sampling rate of 250Hz.

Six sensors were glued on different articulators, viz. Upper Lip (UL), Lower Lip (LP), Jaw, Tongue Tip (TT), Tongue Body (TB), and Tongue Dorsum (TD). Additionally, two sensors were glued behind the ears for head movement correction. In this work, only the movements in the midsagittal plane are considered, corresponding to horizontal and vertical directions. Thus, we have twelve articulatory trajectories denoted by UL_x, UL_y, LL_x, LL_y, Jaw_x, Jaw_y, TT_x, TT_y, TB_x, TB_y, TD_x, TD_y. Manual annotations were done to remove silence at the start and end of each sentence.

3. Proposed Approach

We train different transformer models for AAI and PTA tasks. While an encoder-decoder structure is sufficient for AAI, additional duration modelling is required for PTA. In this section, we briefly summarise the transformer architecture and then describe the details involved in both tasks. A detailed version of the same is made available at https://arxiv.org/abs/2104.05017.

3.1. Transformers

The building block of a transformer layer consists of self-attention with residual connection and a 1D convolutional network.

3.1.1. Self-attention and 1D CNN

We compute the self-attention as carried out in [17]. In multi-head attention, the self-attention function is calculated multiple times parallelly. This increases the model's ability to focus on different positions in the sequence and it also gives the attention layer multiple different representation sub-spaces. Following the attention layer, there is a two-layer 1D convolutional network with ReLU activation. Each sub-layer has a residual connection around it and is followed by layer normalization. Self-attention and 1D CNN form a transformer layer as shown in Fig. 1(b). By stacking transformer layers sequentially, we obtain Feed Forward Transformer (FFT), which is used to represent encoder and decoder blocks.

3.1.2. Positional Encoding

Since there is no positional information encoded in the input sequence, positional encoding is combined with the input. This is usually done by adding or concatenating the sinusoidal positional encoding, as introduced in [17]. Another way to introduce positional information is through relative positional embeddings as utilised in [23], [24]. While Sinusoidal encoding act on the entire sequence length, relative encoding allows for a small receptive field over which unique positional embeddings can be learnt.

3.2. AAI with transformers

We use 2 dense layers as an embedding, followed by an encoder-decoder structure consisting of FFT architecture for the AAI task. Although position information is implicitly encoded
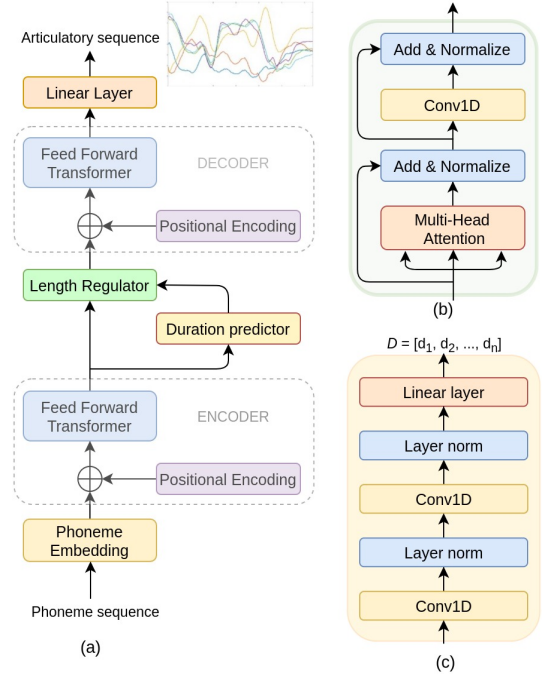

Figure 1: *(a) FastSpeech architecture for PTA task (b) Transformer Layer (c) Duration predictor*

as temporal information in acoustics, it may fail over modelling long term dependencies. Further, with MFCC features as input, adding sinusoidal encoding may harm the model performance as explored in previous work [25]. So we perform AAI experiments with different ways of inserting positional information. We first try with no positional encoding with the assumption that it is present implicitly in MFCCs. We then insert sinusoidal information by adding or concatenating it with the input. Finally, we use relative positional embedding with a total receptive field of 10 time-steps on either side of each position in the sequence.

3.3. PTA with FastSpeech

We use the FastSpeech architecture as shown in Fig. 1(a) for the PTA task. A duration model is necessary for this task since there is no one-to-one correspondence between phonemes and articulator frames. To achieve this, we use a duration predictor with a length regulator.

3.3.1. Duration Predictor

As shown in Fig. 1(c), the duration predictor consists of linear layers which outputs a scalar value for each position in the phoneme sequence. This module is jointly trained with the main model with mean squared error as the loss function.

3.3.2. Length Regulator

Length regulator up-samples the phoneme sequence according to the predicted durations to obtain the length of articulator sequence. For example, let the hidden states of a phoneme sequence be represented as $H_p = [h_1, h_2, h_3, h_4]$, with predicted phoneme durations $D = [2, 2, 3, 1]$, then the expanded sequence becomes $H_p = [h_1, h_1, h_2, h_2, h_3, h_3, h_3, h_4]$.

We use teacher forcing approach during training to obtain the decoder input length directly from ground truth durations.

This is carried out to construct the decoder length accurately from the onset of training which could help in faster loss convergence. This also avoids unnecessary alignment or zero-padding to be performed during training.

During inference, duration predictor outputs are used to construct the decoder input length. The phoneme encoder representation is replicated according to the number of frames to be present for each phoneme, which acts as the input to the decoder. At the output, we remove the zero-padded positions and perform DTW alignment as carried out in [15] using the Euclidean distance metric between the predicted and ground-truth articulatory trajectories and then report the metrics.

3.4. Complexity Analysis

In transformers, the self-attention computation has $O(n^2 d)$ time complexity where n is the sequence length and d is the attention dimension. In LSTMs, it has $O(nd^2)$ complexity on every time-step and $O(n)$ complexity to visit the entire sequence. When n is lesser than d, a transformer layer processes faster than LSTM layer. On the PTA task, the phoneme sequence length at the encoder is 60, while the articulator length at the decoder is fixed at 400 for training. Due to the lower sequence length at the encoder, a large gain can be obtained on training and inference time.

4. Experimental Setup

4.1. Preprocessing

The articulatory data is passed through a low-pass filter, with a cut-off frequency 25Hz, to remove high-frequency noise incurred during the measurement process. This also preserves the smoothly varying nature of the articulatory movement. We then perform sentence-wise mean removal and variance normalisation along each articulator dimension. The recorded speech is down-sampled from 48kHz to 16kHz, following which a forced alignment is performed to obtain phoneme sequence along with their boundaries using Kaldi [26] for every sentence. The resultant phonetic transcription of the dataset consists of 39 ARPABET symbols. We represent the phoneme input as a 1-dimensional vector with indices corresponding to the phonemes present in the sentence. We then zero-pad the vectors to obtain a fixed length equal to the maximum phoneme sequence length, which is set to 60. To represent acoustic data, we use 13-dimensional MFCC features. We zero-pad MFCCs and articulatory data to obtain a fixed length equal to the maximum number of frames, which is set to 400.

4.2. Model training and evaluation

We perform both the PTA and AAI experiments on 10 subjects' data with evaluation on unseen sentences. Each subject has 460 sentences from which 80% of the sentences are used for training, 10% for validation and 10% for testing. We use root mean squared error (RMSE) and correlation coefficient (CC) [27] as evaluation metrics to assess the performance of articulatory movement prediction. These are computed separately across each articulatory dimension and the average values across articulators and subjects are reported.

We examine the performance in 3 types of experimental setups namely subject-dependent (E1), pooled (E2) and fine-tuned (E3). In subject-dependent (E1) case, a model is trained for each subject and evaluated on that particular subject. In the pooled (E2) case, all subjects' data are combined to train a sin-

gle model and this is evaluated across all the subjects. In the case of fine-tuning (E3), we start with the pooled model as a pre-trained model and retrain for each subject separately and follow the evaluation as in E1.

All experiments were performed using PyTorch [28] on a single RTX 2080 GPU. We use NVIDIA's open source code repository for FastPitch [20] implementation as a starting point for the model architectures.

5. Results and Discussion

We evaluate the performance of the proposed approach on AAI and PTA tasks using different experimental setups. We first present the results of the AAI task and then analyse the explicit duration modelling and predicted articulators from the PTA task. We also explore the benefits of having transformer models as representation blocks.

5.1. Performance on AAI

We train transformer models for AAI with different positional encoding, as shown in Table 1. We observe degradation in the performance using additive sinusoidal positional information. We obtain the best performance of 0.885 CC by using relative encoding in the case of E3, though there is a drop in performance in E1 using relative encoding compared to concatenative and no encoding models. We conclude that not using any encoding or using concatenative sinusoidal encoding works well generally when the amount of training data is not large. However, while relative encoding is useful with a large amount of data.

Table 1: *AAI performance using the different models on various positional encodings. Standard deviation across all test sentences is reported in brackets*

Setup	None		Additive		Concatenative		Relative	
	CC	RMSE	CC	RMSE	CC	RMSE	CC	RMSE
E1	0.854 (0.03)	1.062 (0.08)	0.839 (0.03)	1.115 (0.09)	**0.857 (0.02)**	1.060 (0.08)	0.836 (0.02)	1.351 (0.11)
E2	0.871 (0.02)	1.017 (0.08)	0.867 (0.02)	1.038 (0.08)	0.874 (0.02)	1.013 (0.08)	**0.884 (0.02)**	0.971 (0.08)
E3	0.877 (0.02)	0.999 (0.09)	0.874 (0.02)	1.018 (0.09)	0.881 (0.02)	0.996 (0.08)	**0.885 (0.02)**	0.968 (0.09)

In Table 2, we further compare AAI results using BiLSTM and transformers with concatenative encoding. We observe 1.5%, 3% and 3.1% relative improvement in CC using transformers over BiLSTM in subject-dependent (E1), pooled (E2) and fine-tuned (E3) setups respectively. We observe that, although transformers have lesser parameters (Table 3), it performs better than BiLSTM on the AAI task.

5.2. Performance on PTA

For the PTA task, we use the FastSpeech architecture with the duration predictor module. As shown in Table 2, FastSpeech performs better than Tacotron in E1 setup, improving CC by 154% (relative). This could mainly be due to the hard alignment obtained with the duration predictor allowing it to reliably predict articulator movements even with less data. On the other hand, the attention-based duration modelling in Tacotron needs more data to learn alignments. We apply the same duration predictor module on BiLSTM as encoder and decoder instead of transformers. However, it fails to learn good representation with BiLSTM as an encoder. Our experiments suggest that trans-

formers are more compatible with the explicit duration predictor on the intermediate representation between the encoder and decoder.

Table 2: *PTA and AAI performance using different models in different training setups. Standard deviation across all test sentences is reported in brackets*

Setup	PTA				AAI			
	Tacotron		FastSpeech		BiLSTM		Tranformers	
	CC	RMSE	CC	RMSE	CC	RMSE	CC	RMSE
E1	0.329	2.045	**0.838**	1.162	0.844	1.05	**0.857**	1.06
	(0.03)	(0.1)	**(0.03)**	(0.08)	(0.02)	(0.08)	**(0.03)**	(0.08)
E2	0.682	1.482	**0.763**	1.486	0.849	1.01	**0.875**	1.013
	(0.04)	(0.09)	**(0.05)**	(0.18)	(0.02)	(0.08)	**(0.02)**	(0.08)
E3	0.806	1.18	**0.845**	1.124	0.854	0.99	**0.881**	0.996
	(0.03)	(0.08)	**(0.03)**	(0.08)	(0.02)	(0.085)	**(0.02)**	(0.08)

In the E2 setup, we obtain 11.8% improvement in CC over the Tacotron. We utilise these pooled models to fine-tune each subject. In E3, we observe an improvement of 4.8% in CC. FastSpeech performs better than the Tacotron in all experimental setups although they have identical model size (Table 3).

In Fig. 2, we analyse the predicted durations from FastSpeech and Tacotron models in E3 setup on 10 phonemes. It is clear that the predicted duration in the case of subject M1 (Fig. 2(a)) are not significantly ($p > 0.05$) different than ground truth duration except for /ae/ & /ah/ for FastSpeech and /th/ for Tacotron. With subject F2 (Fig. 2(b)), a significant difference ($p < 0.05$) between the ground truth and predicted duration is observed only in the case of /dh/ and /m/ for FastSpeech and Tacotron respectively. We observe that ground truth and predicted durations matches in the majority of the cases using Tacotron and FastSpeech on fine-tuned (E3) setup.
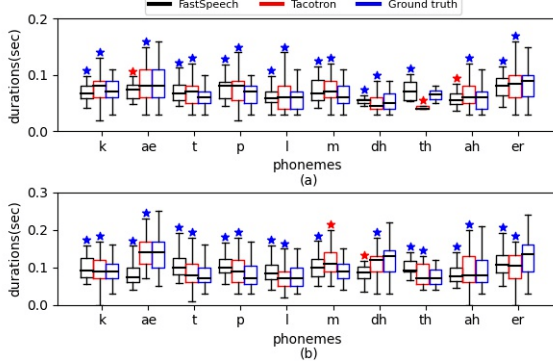

Figure 2: *Ground truth vs predicted durations for 10 phonemes for subjects (a) M1 and (b) F2. ⋆ indicates the cases where the predicted duration is not statistically significantly ($p > 0.05$) different from the ground truth duration. ⋆ indicates cases where they are significantly different ($p < 0.05$)*

In Fig. 3, we compare LL_y and TT_y ground truth articulatory trajectories along with FastSpeech and Tacotron predictions. On the ground truth articulatory movements, we mark phoneme boundaries obtained by forced alignment. We extract the predicted durations from FastSpeech and show it with the predicted articulators. For Tacotron, we use the attention alignment distribution from the trained network. We observe that FastSpeech predictions closely resemble the ground truth, both in phoneme durations and the articulatory trajectories. For example, consider $t = 0.1$ when phoneme /ih/ is uttered, a dip is observed in LL_y curve in ground truth and the FastSpeech predictions while it is flatter in the Tacotron prediction.
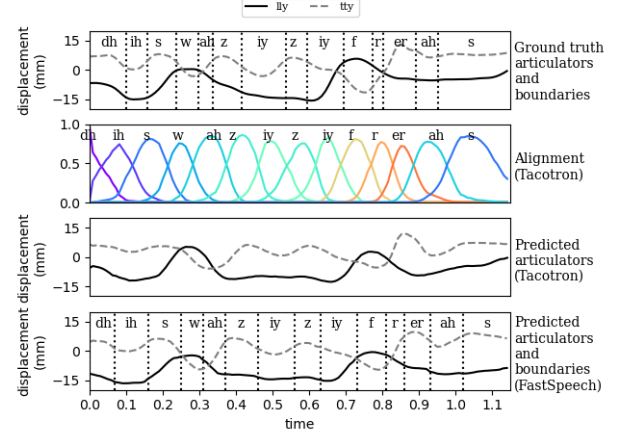

Figure 3: *Comparing original LL_y and TT_y with the estimated ones from phonemes*

5.3. Time-memory usage

We further compare the time and memory usage of the different models in Table 3. On the AAI task, we use BiLSTM with 3.7M parameters and a transformer Encoder-Decoder with 1.67M parameters. We observe transformer and BiLSTM have similar training time in AAI task. For the PTA task, we fix the number of model parameters at 27M for FastSpeech and Tacotron. We notice that FastSpeech is 6 times faster in E1 setup and 37 times faster in E2 setup training. This is possible due to significant parallelization on how representation is learnt across a sequence. We also observe a lot of speedup in inference time with FastSpeech. From these observations, we can conclude that transformer architectures are better suited for PTA task. On comparing the transformer models on the two tasks, we notice that AAI uses 40 times smaller model compared to the PTA task, further work is required to bring down the model size of PTA models.

Table 3: *Time and memory usage of different models*

Setup	PTA		AAI	
	FastSpeech	Tacotron	Tranformers	BiLSTM
E1 train time	12 mins	74 mins	7 mins	3 mins
E2 train time	21 mins	13 hrs	45 mins	1 hr
Inference time	0.007 sec	0.44 sec	0.003 sec	0.0006 sec
No of parameters	27M	27M	1.67M	3.71M

6. Conclusions

In this work, we propose the benefit of transformer based networks for the estimation of articulatory movements. We show relative improvement of 154%, 11.8% and 4.8% in CC on subject-dependent (E1), pooled (E2) and fine-tuned (E3) setups respectively on the PTA task. We also show that having an explicit duration predictor is beneficial towards obtaining hard alignment with the articulator movements. This works particularly well on limited data scenarios. With transformer models for AAI, we improve the CC by 1.5%, 3% and 3.1% on the same setups. We conclude that transformer network is a good alternative to learn articulator movements from phonemes and acoustics. We also observe that articulatory movements are predicted more accurately from acoustics (AAI) compared to phonemes (PTA). This is consistent with the findings from the past. In future, we plan on improving PTA to attain performance on par with AAI.

7. References

[1] J. Frankel, K. Richmond, S. King, and P. Taylor, "An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces," *Proceedings of the International Conference on Spoken Language Processing*, Beijing, China (CD-ROM) 2000.

[2] K. Kirchhoff, "Robust speech recognition using articulatory information," Ph.D. dissertation, University of Bielefeld, 1999.

[3] C. S, C. Yarra, R. Aggarwal, S. K. Mittal, K. N K, R. K T, A. Singh, and P. K. Ghosh, "Automatic visual augmentation for concatenation based synthesized articulatory videos from real-time MRI data for spoken language training," in *Proc. Interspeech*, 2018, pp. 3127–3131.

[4] U. Desai, C. Yarra, and P. K. Ghosh, "Concatenative articulatory video synthesis using real-time MRI data for spoken language training," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4999–5003.

[5] A. Illa and P. K. Ghosh, "An investigation on speaker specific articulatory synthesis with speaker independent articulatory inversion," in *Proc. Interspeech*, 2019, pp. 121–125. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2664

[6] Z.-H. Ling, K. Richmond, J. Yamagishi, and R.-H. Wang, "Integrating articulatory features into HMM-based parametric speech synthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1171–1185, 2009.

[7] T. Toda, A. W. Black, and K. Tokuda, "Statistical mapping between articulatory movements and acoustic spectrum using a Gaussian mixture model," *Speech Communication*, vol. 50, no. 3, pp. 215–227, 2008.

[8] L. Zhang and S. Renals, "Acoustic-articulatory modeling with the trajectory HMM," *IEEE Signal Processing Letters*, vol. 15, pp. 245–248, 2008.

[9] K. Richmond, "A trajectory mixture density network for the acoustic-articulatory inversion mapping." in *Proceedings of the ICSLP, Pittsburgh*, 2006, pp. 577–580.

[10] Z. Wu, K. Zhao, X. Wu, X. Lan, and H. Meng, "Acoustic to articulatory mapping with deep neural network," *Multimedia Tools and Applications*, vol. 74, no. 22, pp. 9889–9907, 2015.

[11] A. Illa and P. K. Ghosh, "Low resource acoustic-to-articulatory inversion using bi-directional long short term memory," in *Proc. Interspeech*, 2018, pp. 3122–3126.

[12] P. Liu, Q. Yu, Z. Wu, S. Kang, H. Meng, and L. Cai, "A deep recurrent approach for acoustic-to-articulatory inversion," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4450–4454.

[13] Z.-H. Ling, K. Richmond, and J. Yamagishi, "An analysis of HMM-based prediction of articulatory movements," *Speech Communication*, vol. 52, no. 10, pp. 834–846, 2010.

[14] P. Zhu, L. Xie, and Y. Chen, "Articulatory movement prediction using deep bidirectional long short-term memory based recurrent neural networks and word/phone embeddings," in *INTERSPEECH, Dresden, Germany, September 6-10*, 2015, pp. 2192–2196. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2015/i15_2192.html

[15] A. Singh, A. Illa, and P. K. Ghosh, "A comparative study of estimating articulatory movements from phoneme sequences and acoustic features," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7334–7338.

[16] J. Shen *et al.*, "Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4779–4783.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[18] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, "Neural speech synthesis with transformer network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 6706–6713, 07 2019.

[19] Y. Ren, Y. Ruan, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu, "Fastspeech: Fast, robust and controllable text to speech," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.

[20] A. Łańcucki, "FastPitch: Parallel Text-to-speech with Pitch Prediction," *arXiv e-prints*, p. arXiv:2006.06873, Jun. 2020.

[21] "EM9600 shotgun microphone," avaliable online: http://www.tbone-mics.com/en/product/information/details/the-tbone-em-9600-richtrohr-mikrofon/, last accessed:4/2/2020.

[22] "3d electromagnetic articulograph," available online: http://www.articulograph.de/, last accessed: 4/2/2020.

[23] P. Shaw, J. Uszkoreit, and A. Vaswani, "Self-attention with relative position representations," *CoRR*, vol. abs/1803.02155, 2018.

[24] P. Zhou, R. Fan, W. Chen, and J. Jia, "Improving generalization of transformer for speech recognition with parallel schedule sampling and relative positional embedding," *CoRR*, vol. abs/1911.00203, 2019. [Online]. Available: http://arxiv.org/abs/1911.00203

[25] M. Sperber, J. Niehues, G. Neubig, S. Stüker, and A. Waibel, "Self-Attentional Acoustic Models," *arXiv e-prints*, p. arXiv:1803.09519, Mar. 2018.

[26] D. Povey et al., "The Kaldi speech recognition toolkit," in *IEEE workshop on automatic speech recognition and understanding*, 2011.

[27] P. K. Ghosh and S. Narayanan, "A generalized smoothness criterion for acoustic-to-articulatory inversion," *The Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 2162–2172, 2010. [Online]. Available: https://asa.scitation.org/doi/abs/10.1121/1.3455847

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.