An Improved Model for Voicing Silent Speech

David Gaddy and **Dan Klein**
University of California, Berkeley
{dgaddy,klein}@berkeley.edu

Abstract

In this paper, we present an improved model for voicing silent speech, where audio is synthesized from facial electromyography (EMG) signals. To give our model greater flexibility to learn its own input features, we directly use EMG signals as input in the place of hand-designed features used by prior work. Our model uses convolutional layers to extract features from the signals and Transformer layers to propagate information across longer distances. To provide better signal for learning, we also introduce an auxiliary task of predicting phoneme labels in addition to predicting speech audio features. On an open vocabulary intelligibility evaluation, our model improves the state of the art for this task by an absolute 25.8%.

1 Introduction

EMG-based voicing of silent speech is a task that aims to synthesize vocal audio from muscular signals captured by electrodes on the face while words are silently mouthed (Gaddy and Klein, 2020; Toth et al., 2009). While recent work has demonstrated a high intelligibility of generated audio when restricted to a narrow vocabulary (Gaddy and Klein, 2020), in a more challenging open vocabulary setting the intelligibility remained low (68% WER). In this work, we introduce an new model for voicing silent speech that greatly improves intelligibility.

We achieve our improvements by modifying several different components of the model. First, we improve the input representation. While prior work on EMG speech processing uses hand-designed features (Jou et al., 2006; Diener et al., 2015; Meltzner et al., 2018; Gaddy and Klein, 2020) which may throw away some information from the raw signals, our model learns directly from the complete signals with minimal pre-processing by using a set of convolutional neural network layers as feature extractors. This modification follows recent work in speech processing from raw waveforms (Collobert et al., 2016; Schneider et al., 2019) and gives our model the ability to learn its own features for EMG.

Second, we improve the neural architecture of the model. While other silent speech models have been based around recurrent layers such as LSTMs (Janke and Diener, 2017; Gaddy and Klein, 2020), we use the self-attention-based Transformer architecture (Vaswani et al., 2017), which has been shown to be a more powerful replacement across a range of tasks.

Finally, we improve the signal used for learning. Since the relatively small data sizes for this task creates a challenging learning problem, we introduce an auxiliary task of predicting phoneme labels to provide additional guidance. This auxiliary loss is inspired by prior work on the related problem of generating speech from ECoG sensors on the brain, which greatly benefited from intermediate prediction of phonemic information (Anumanchipalli et al., 2019).

We evaluate intelligibility of audio synthesized by our model on the single-speaker data from Gaddy and Klein (2020) in the most challenging open-vocabulary setting. Our results reflect an absolute improvement in error rate of 25.8% over the state of the art, from 68.0% to 42.2%, as measured by automatic transcription. Evaluation by human transcription gives an even lower error rate of 32%.

2 Model

At a high level, our system works by predicting a sequence of speech features from EMG signals and using a WaveNet vocoder (van den Oord et al., 2016) to synthesize audio from those predicted features, as was done in Gaddy and Klein (2020). The first component, dubbed the transduction model, takes in EMG signals from eight electrodes around

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 175–181
August 1–6, 2021. ©2021 Association for Computational Linguistics

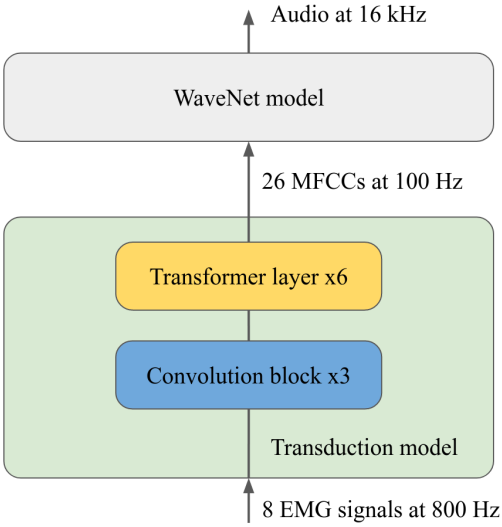

Figure 1: Model overview

the face and outputs a sequence of speech features represented as Mel-frequency cepstral coefficients (MFCCs). The final step of vocoding audio from MFCCs is unchanged in our work, so we defer to Gaddy and Klein (2020) for the details of the WaveNet model.

The neural architecture for our transduction model is made up of a set of residual convolution blocks followed by a transformer with relative position embeddings, as shown in Figure 1. We describe these two components in Sections 2.1 and 2.2 below. Next, in Section 2.3 we describe our training procedure, which aligns each silent utterance to a corresponding vocalized utterance as in Gaddy and Klein (2020) but with some minor modifications. Finally, in Section 2.4 we describe the auxiliary phoneme-prediction loss that provides additional signal to our model during training.[1]

2.1 Convolutional EMG Feature Extraction

The convolutional layers of our model are designed to directly take in EMG signals with minimal preprocessing. Prior to use of the input EMG signals, AC electrical noise is removed using band stop filters at harmonics of 60 Hz, and DC offset and drift are removed with a 2 Hz high-pass filter. The signals are then resampled from 1000 Hz to 800 Hz, and the magnitudes are scaled down by a factor of 10.

Our convolutional architecture uses a stack of 3 residual convolution blocks inspired by ResNet (He et al., 2016), but modified to use 1-dimensional

[1]Code for our model is available at https://github.com/dgaddy/silent_speech.

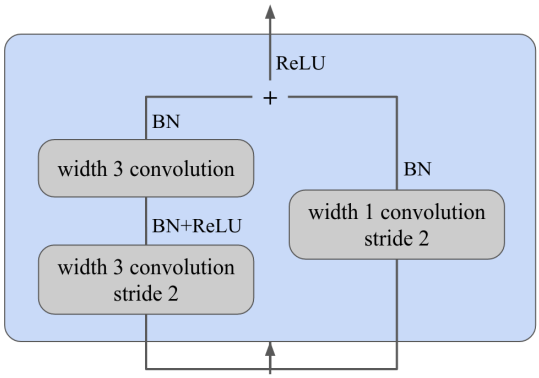

Figure 2: Convolution block architecture

convolutions. The architecture used for each convolution block is shown in Figure 2, and has two convolution-over-time layers along the main path as well as a shortcut path that does not do any aggregation over the time dimension. Each block downsamples the signal by a factor of 2, so that the input signals at 800 Hz are eventually transformed into features at 100Hz to match the target speech feature frame rate. All convolutions have channel dimension 768.

Before passing the convolution layer outputs to the rest of the model, we include an embedding of the session index, which helps the model account for differences in electrode placement after electrodes are reattached for each session. Each session is represented with a 32 dimensional embedding, which is projected up to 768 dimensions with a linear layer before adding to the convolution layer outputs at each timestep.

2.2 Transformer with Relative Position Embeddings

To allow information to flow across longer time horizons, we use a set of bidirectional Transformer encoder layers (Vaswani et al., 2017) on top of the convolution layers in our model. To capture the time-invariant nature of the task, we use relative position embeddings as described by Shaw et al. (2018) rather than absolute position embeddings. In this variant, a learned vector p that depends on the relative distance between the query and key positions is added to the key vectors when computing attention weights. Thus, the attention logits are computed with

$$e_{ij} = \frac{(W_K x_j + p_{ij})^\top (W_Q x_i)}{\sqrt{d}}$$

where p_{ij} is an embedding lookup with index $i - j$, up to a maximum distance k in each direction (x are inputs to the attention module, W_Q and W_K are query and key transformations, and d is the dimension of the projected vectors $W_Q x_i$). For our model, we use $k = 100$ (giving each layer a 1 second view in each direction) and set all attention weights with distance greater than k to zero. We use six of these Transformer layers, with 8 heads, model dimension 768, feedforward dimension 3072, and dropout 0.1.

The output of the last Transformer layer is passed through a final linear projection down to 26 dimensions to give the MFCC audio feature predictions output by the model.

2.3 Alignment and Training

Since silent EMG signals and vocalized audio features must be recorded separately and so are not time-aligned, we must form an alignment between the two recordings to calculate a loss on predictions from silent EMG. Our alignment procedure is similar to the predicted-audio loss used in Gaddy and Klein (2020), but with some minor aspects improved.

Our loss calculation takes in a sequence of MFCC features $\hat{A}_S$ predicted from silent EMG and another sequence of target features A_V from a recording of vocalized audio for the same utterance. We compute a pairwise distance between all pairs of features

$$\delta[i, j] = \left\| A_V[i] - \hat{A}_S[j] \right\|_2$$

and run dynamic time warping (Rabiner and Juang, 1993) to find a minimum-cost monotonic alignment path through the δ matrix. We represent the alignment as $a[i] \rightarrow j$ with a single position j in $\hat{A}_S$ for every index i in A_V, and take the first such position when multiple are given by dynamic time warping. The loss is then the mean of aligned pairwise distances:

$$L = \frac{1}{N_V} \sum_{i=1}^{N_V} \delta[i, a[i]]$$

In addition to the silent-EMG training, we also make use of EMG recordings during vocalized speech which are included in the data from Gaddy and Klein (2020). Since the EMG and audio targets are recorded simultaneously for these vocalized examples, we can calculate the pairwise distance loss directly without any dynamic time warping. We train on the two speaking modes simultaneously.

To perform batching across sequences of different lengths during training, we concatenate a batch of EMG signals across time then reshape to a batch of fixed-length sequences before feeding into the network. Thus if the fixed batch-sequence-length is l, the sum of sample lengths across the batch is N_S, and the signal has c channels, we reshape the inputs to size $(\lceil N_S / l \rceil, l, c)$ after zero-padding the concatenated signal to a multiple of l. After running the network to get predicted audio features, we do the reverse of this process to get a set of variable-length sequences to feed into the alignment and loss described above. This batching strategy allows us to make efficient use of compute resources and also acts as a form of dropout regularization where slicing removes parts of the nearby input sequence. We use a sequence length $l = 1600$ (2 seconds) and select batches dynamically up to a total length of $N_{Smax} = 204800$ samples (256 seconds).

We train our model for 80 epochs using the AdamW optimizer (Loshchilov and Hutter, 2017). The peak learning rate is 10^{-3} with a linear warm-up of 500 batches, and the learning rate is decayed by half after 5 consecutive epochs of no improvement in validation loss. Weight decay 10^{-7} is used for regularization.

2.4 Auxiliary Phoneme Loss

To provide our model with additional training signal and regularize our learned representations, we introduce an auxiliary loss of predicting phoneme labels at each output frame.

To get phoneme labels for each feature frame of the vocalized audio, we use the Montreal Forced Aligner (McAuliffe et al., 2017). The aligner uses an acoustic model trained on the LibriSpeech dataset in conjunction with a phonemic dictionary to get time-aligned phoneme labels from audio and a transcription.

We add an additional linear prediction layer and softmax on top of the Transformer encoder to predict a distribution over phonemes. For training, we modify the alignment and loss cost δ by appending a term for phoneme negative log likelihood:

$$\delta'[i, j] = \left\| A_V[i] - \hat{A}_S[j] \right\|_2 - \lambda P_V[i]^\top \log \hat{P}_S[j]$$

where $\hat{P}_S$ is the predicted distribution from the model softmax and P_V is a one-hot vector for the target phoneme label. We use $\lambda = .1$ for the phoneme loss weight. After training, the phoneme prediction layer is discarded.

Model	WER
Gaddy and Klein (2020)	68.0
This work	**42.2**
Ablation: Replace convolution features with hand-designed features	45.2
Ablation: Replace Transformer with LSTM	46.0
Ablation: Remove phoneme loss	51.7

Table 1: Open vocabulary word error rate results from an automatic intelligibility evaluation.

3 Results

We train our model on the open-vocabulary data from Gaddy and Klein (2020). This data contains 19 hours of facial EMG data recordings from a single English speaker during silent and vocalized speech. Our primary evaluation uses the automatic metric from that work, which transcribes outputs with an automatic speech recognizer[2] and compares to a reference with a word error rate (WER) metric. We also evaluate human intelligibility in Section 3.1 below.[3]

The results of the automatic evaluation are shown in Table 1. Overall, we see that our model improves intelligibility over prior work by an absolute 25.8%, or 38% relative error reduction. Also shown in the table are ablations of our three primary contributions. We ablate the convolutional feature extraction by replacing those layers with the hand-designed features used in Gaddy and Klein (2020), and we ablate the Transformer layers by replacing with LSTM layers in the same configuration as that work (3 bidirectional layers, 1024 dimensions). To ablate the phoneme loss, we simply set its weight in the overall loss to zero. All three of these ablations show an impact on our model's results.

3.1 Human Evaluation

In addition to the automatic evaluation, we performed a human intelligibility evaluation using a similar transcription test. Two human evaluators without prior knowledge of the text were asked to listen to 40 synthesized samples and write down the words they heard (see Appendix A for full instructions given to evaluators). We then compared these transcriptions to the ground-truth reference with a WER metric.

The resulting word error rates from the two human evaluators' transcriptions are 36.1% and 28.5% (average: 32.3%), compared to 42.2% from automatic transcriptions. These results validate the improvement shown in the automatic metric, and indicate that the automatic metric may be underestimating intelligibility to humans. However, the large variance across evaluators shows that the automatic metric may still be more appropriate for establishing consistent evaluations across different work on this task.

4 Phoneme Error Analysis

One additional advantage to using an auxiliary phoneme prediction task is that it provides a more easily interpretable view of model predictions. Although the phoneme predictions are not directly part of the audio synthesis process, we have observed that mistakes in audio and phoneme prediction are often correlated. Therefore, to better understand the errors that our model makes, we analyze the errors of our model's phoneme predictions. To analyze the phoneme predictions, we align predictions on a silent utterance to phoneme labels of a vocalized utterance using the procedure described above in Sections 2.3 and 2.4, then evaluate the phonemes using the measures described in Sections 4.1 and 4.2 below.

4.1 Confusion

First, we measure the confusion between each pair of phonemes. We use a frequency-normalized metric for confusion: $(e_{p1,p2} + e_{p2,p1})/(f_{p1} + f_{p2})$, where $e_{p1,p2}$ is the number of times $p2$ was predicted when the label was $p1$, and f_{p1} is the number of times phoneme $p1$ appears as a target label. Figure 3 illustrates this measure of confusion using darkness of lines between the phonemes, and Appendix B lists the values of the most confused pairs.

We observe that many of the confusions are be-

[2] An implementation of DeepSpeech (Hannun et al., 2014) from Mozilla (https://github.com/mozilla/DeepSpeech)

[3] Output audio samples available at https://dgaddy.github.io/silent_speech_samples/ACL2021/.

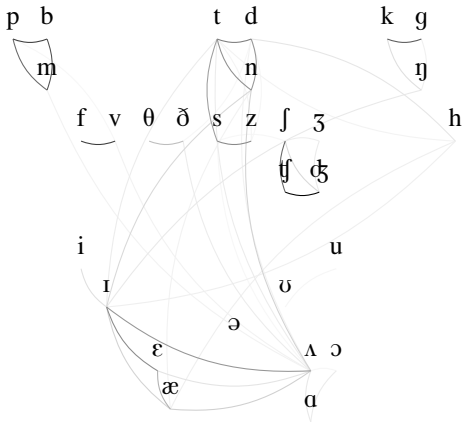

Figure 3: Phoneme confusability (darker lines indicate more confusion - maximum darkness is 13% confusion)

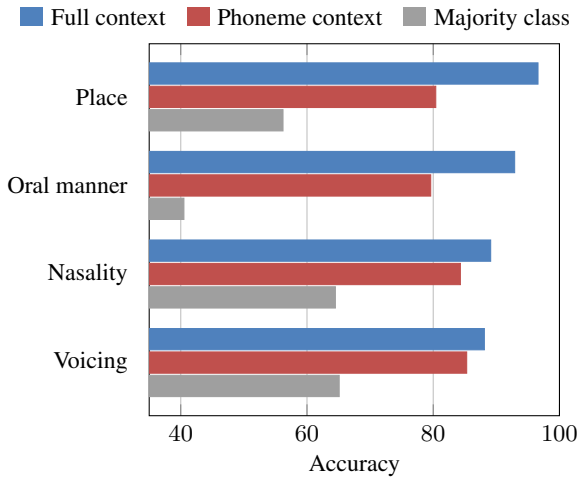

Figure 4: Accuracy of selecting phonemes along articulatory feature dimensions. We compare our full EMG model (full context) with a majority class baseline and a model given only phoneme context as input.

tween pairs of consonants that differ only in voicing, which is consistent with the observation in Gaddy and Klein (2020) that voicing signals appear to be subdued in silent speech. Another finding is a confusion between nasals and stops, which is challenging due to the role of the velum and its relatively large distance from the surface electrodes, as has been noted in prior work (Freitas et al., 2014). We also see some confusion between vowel pairs and between vowels and consonants, though these patterns tend to be less interpretable.

4.2 Articulatory Feature Accuracy

To better understand our model's accuracy across different consonant articulatory features, we perform an additional analysis of phoneme selection across specific feature dimensions. For this analysis, we define a confusion set for an articulatory feature as a set of English phonemes that are identical across all other features. For example, one of the confusion sets for the place feature is $\{p, t, k\}$, since these phonemes differ in place of articulation but are the same along other axes like manner and voicing (a full listing of confusion sets can be found in Appendix C). For each feature of interest, we calculate a forced-choice accuracy within the confusion sets for that feature. More specifically, we find all time steps in the target sequence with labels belonging in a confusion set and restrict our model output to be within the corresponding set for those positions. We then compute an accuracy across all those positions that have a confusion set.

To evaluate how much of the articulatory feature accuracies can be attributed to contextual inferences rather than information extracted from EMG, we compare our results to a baseline model that is trained to make decisions for a feature based on nearby phonemes. In the place of EMG feature inputs, this baseline model is given the sequence of phonemes predicted by the full model, but with information about the specific feature being tested removed by collapsing phonemes in each of its confusion sets to a single symbol. Additional details on this baseline model can be found in Appendix C.

The results of this analysis are shown in Figure 4. By comparing the gap in accuracy between the full model and the phoneme context baseline, we again observe trends that correspond to our prior expectations. While place and oral manner features can be predicted much better by our EMG model than from phonemic context alone, nasality and voicing are more challenging and have a smaller improvement over the contextual baseline.

5 Conclusion

By improving several model components for voicing silent speech, our work has achieved a 38% relative error reduction on this task. Although the problem is still far from solved, we believe the large rate of improvement is a promising sign for continued progress.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1618460 and by DARPA under the LwLL program / Grant No. FA8750-19-1-0504.

References

Gopala K Anumanchipalli, Josh Chartier, and Edward F Chang. 2019. Speech synthesis from neural decoding of spoken sentences. *Nature*, 568(7753):493–498.

Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193*.

Lorenz Diener, Matthias Janke, and Tanja Schultz. 2015. Direct conversion from facial myoelectric signals to speech using deep neural networks. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE.

João Freitas, António JS Teixeira, Samuel S Silva, Catarina Oliveira, and Miguel Sales Dias. 2014. Velum movement detection based on surface electromyography for speech interface. In *BIOSIGNALS*, pages 13–20.

David Gaddy and Dan Klein. 2020. Digital voicing of silent speech. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5521–5530, Online. Association for Computational Linguistics.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Matthias Janke and Lorenz Diener. 2017. EMG-to-speech: Direct generation of speech from facial electromyographic signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(12):2375–2385.

Szu-Chen Jou, Tanja Schultz, Matthias Walliczek, Florian Kraft, and Alex Waibel. 2006. Towards continuous speech recognition using surface electromyography. In *Ninth International Conference on Spoken Language Processing*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. 2017. Montreal forced aligner: Trainable text-speech alignment using kaldi. In *Interspeech*, volume 2017, pages 498–502.

Geoffrey S Meltzner, James T Heaton, Yunbin Deng, Gianluca De Luca, Serge H Roy, and Joshua C Kline. 2018. Development of sEMG sensors and algorithms for silent speech recognition. *Journal of neural engineering*, 15(4):046031.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *ArXiv*, abs/1609.03499.

Lawrence Rabiner and Biing-Hwang Juang. 1993. *Fundamentals of speech recognition*. Prentice Hall.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *INTERSPEECH*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Arthur R. Toth, Michael Wand, and Tanja Schultz. 2009. Synthesizing speech from electromyography using voice transformation techniques. In *INTERSPEECH*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

A Instructions to Human Evaluators

The following instructions were given to human evaluators for the transcription test described in Section 3.1:

Please listen to each of the attached sound files and write down what you hear. There are 40 files, each of which will contain a sentence in English. Write your transcriptions into a spreadsheet such as Excel or Google sheets so that the row numbers match the numbers in the file names. Many of the clips may be difficult to hear. If this is the case, write whatever words you are able to make out, even if it does not form a complete expression. If you are not entirely sure about a word but can make a strong guess, you may include it in your transcription, but only do so if you beleive it is more likely than not to be the correct word. If you cannot make out any words, leave the corresponding row blank.

B Phoneme Confusability

This section provides numerical results for phoneme confusions to complement the illustration given in Section 4.1 of the main paper. We compare the frequency of errors between two phonemes to the frequency of correct predictions on those phonemes. We define the following two quantities:

$$\text{Confusion: } (e_{p1,p2} + e_{p2,p1})/(f_{p1} + f_{p2})$$

$$\text{Accuracy: } (e_{p1,p1} + e_{p2,p2})/(f_{p1} + f_{p2})$$

where $e_{p1,p2}$ is the number of times $p2$ was predicted when the label was $p1$, and f_{p1} is the number of times phoneme $p1$ appears as a target label. Results for the most confused pairs are shown in the table below.

Phonemes		Confusion (%)	Accuracy (%)
ʤ	ʧ	13.2	49.4
v	f	10.4	72.0
p	b	10.3	64.3
m	b	9.3	74.3
k	ɡ	8.9	77.2
ʃ	ʧ	8.3	59.8
p	m	8.1	73.0
t	d	7.2	64.0
z	s	6.6	80.0
ɪ	ɛ	6.5	60.6
t	n	6.3	67.1
n	d	6.0	66.8
ɪ	ʌ	6.0	65.8
ɹ	ɚ	5.7	78.2
t	s	5.5	72.8
ɛ	æ	4.7	70.9
u	oʊ	4.3	77.4
θ	ð	4.1	76.9
ʌ	æ	3.2	72.1
ɪ	æ	3.1	64.9

C Articulatory Feature Analysis Details

The following table lists all confusion sets used in our articulatory feature analysis in Section 4.2.

Feature	Confusion Sets
Place	{p,t,k} {b,d,g} {m,n,ŋ} {f,θ,s,ʃ,h} {v,ð,z,ʒ}
Oral manner	{t,s} {d,z,l,r} {ʃ,ʧ} {ʒ,ʤ}
Nasality	{b,m} {d,n} {g,ŋ}
Voicing	{p,b} {t,d} {k,g} {f,v} {θ,ð} {s,z} {ʃ,ʒ} {ʧ,ʤ}

The phoneme context baseline model uses a Transformer architecture with dimensions identical to our primary EMG-based model, but is fed phoneme embeddings of dimension 768 in the place of the convolutional EMG features. The phonemes input to this model are the maximum-probability predictions output by our primary model at each frame, but with all phonemes from a confusion set replaced with the same symbol. We train a separate baseline model for each of the four articulatory feature types to account for different collapsed sets in the input. During training, a phoneme likelihood loss is applied to all positions and no restrictions are enforced on the output. Other training hyperparameters are the same between this baseline and the main model.

D Additional Reproducibility Information

All experiments were run on a single Quadro RTX 6000 GPU, and each took approximately 12 hours. Hyperparameters were tuned manually based on automatic transcription WER on the validation set. The phoneme loss weight hyperparameter λ was chosen from $\{1, .5, .1, .05, .01, .005\}$. We report numbers on the same test split as Gaddy and Klein (2020), but increase the size of the validation set to 200 examples to decrease variance during model exploration and tuning. Our model contains approximately 40 million parameters.