Adapting Listen, Attend, and Spell to Enhance Brain-Computer Interfaces for Speech Decoding

Stanford CS224N Custom Project

Dylan Iskandar
Department of Computer Science
Stanford University
dylanisk@stanford.edu

Brian Ni
Department of Computer Science
Stanford University
brianni@stanford.edu

Vedant Singh
Department of Mathematics
Stanford University
vedants@stanford.edu

Abstract

Our goal is to improve brain-computer interfaces (BCIs) for individuals with speech impairments by adapting the Listen, Attend, and Spell (LAS) model (Chan et al., 2015) to process neural signals. We use the same dataset as the speechBCI model (Willett et al., 2023a), which contains the neural activity that we decode into text. However, we propose two variants of the LAS model: (1) Phoneme Prediction, configured to predict phonemes, the basic units of sound in speech, and (2) Word/Subword Prediction, designed to predict words or subword tokens. For the Word/Subword Prediction variant, we use SentencePiece and Byte Pair Encoding (BPE) for subword tokenization to improve decoding accuracy and communication speed. Comparing the Phoneme Prediction variant with our PyTorch baseline in terms of Phoneme Error Rate (PER) and the Word/Subword Prediction variant with our baseline plus Language Models (LMs) in terms of Word Error Rate (WER), we see that the performance of our model is slightly below that of the baseline. We hope to observe significant improvements in decoding accuracy and communication speed in our subsequent results, which can potentially transform the quality of life for individuals with severe speech impairments.

1 Key Information to include

- Mentor: Chaofei Fan

2 Introduction

Neuromuscular diseases such as amyotrophic lateral sclerosis (ALS) often result in paralysis and speech impairment, which can significantly impact quality of life. For such individuals, BCIs offer a promising opportunity to restore communication by translating neural signals directly into speech or text. However, existing BCI models have yet to achieve the accuracy and throughout necessary for seamless, general communication. One bottleneck is the task of accurately decoding neural signals, which are inherently noisy and variable, into coherent speech. While these systems can improve quality of life, most users struggle to achieve communication rates exceeding 10 words per minute, far below the natural speech average of 150 words per minute. Recent attempts (such as the speechBCI model developed by Willett et al. (2023b)) have made significant strides in enhancing the reliability and utility of speech BCIs, but are not yet sufficient to render them capable of supporting rapid communication.

Many existing approaches are component-based, requiring separate training cycles for different model components. This segmented approach introduces unnecessary complexity and limits efficiency, as there is no integrated end-to-end model.

Our paper describes an approach aimed at engineering a faster and more accurate end-to-end speech BCI, decoding neural activity into text using recurrent neural networks (RNNs) and attention-based mechanisms. We propose two variants of the model: one focused on text conversion using phoneme prediction, and the other using word/subword prediction. Our evaluations have shown promising results and encourage further research in the field.

3 Related Work

Sequence-to-sequence (seq2seq) models have been widely utilized across various applications, from neural machine translation to conversational modeling; however, their application to speech recognition, particularly from brain signals to phonemes or speech, remains underexplored. Seq2seq models are well-suited for tasks involving different lengths of input and output, making them an ideal candidate for brain signal to speech recognition tasks.

Significant progress has been made in speech decoding based off of neural activity. For instance, Herff et al. (2015) demonstrated the potential of brain-to-text decoding by translating spoken phrases from phoneme representations in the brain. However, their approach was limited by a restricted vocabulary, which hampers generalizability and scalability.

Consonant-Vowel-Consonant (CVC) Word Structures Another notable study, "Generating Natural, Intelligible Speech from Brain Activity in Motor, Premotor, and Inferior Frontal Cortices," by Herff et al. (2019) focused on specific consonant-vowel-consonant (CVC) word structures. This model, while innovative, is not generalizable due to its narrow applicability. Most other models employ the Connectionist Temporal Classification (CTC) approach to training. However, CTC assumes conditional independence of the output tokens; hence, it overlooks the context-dependent nature of speech and conversation, which can result in lower accuracy.

In contrast, our attention-based model addresses these limitations by avoiding the pitfalls of CTC loss and ensuring context-dependency in training. We propose a generalizable and scalable approach trained and tested on a large corpus. By using an end-to-end model with an attention-based mechanism, our approach simplifies the process and enhances the robustness of neural-to-text decoding.

4 Approach

Our approach is based on an end-to-end framework with attention mechanisms. The input we use—the subject's neural activity—is measured from area 6v and 44 of the brain; this dataset was originally collected for use in the BCI paper (Willett et al., 2023b). We first explain the basic approach, followed by the differences introduced in both of our variant models.

The LAS end-to-end model involves both an encoder and a decoder, accepts brain neural activity as the input, and outputs alphanumeric characters. The encoder is a bidirectional Long Short Term Memory RNN (BLSTM) with a pyramid structure, like the architrcture in Clockwork RNN by Koutník et al. (2014). We have three pyramid BLSTM layers on top of the bottom BLSTM layer, which converts the input sequence x into features h. This reduces the number and length of our input features to a more computationally feasible scale. We reduce the time dimension by a factor of two in each successive layer of the pyramid BLSTM (pBLSTM), thus reducing it by eight times in total over the three layers. In our pBLSTM model, we use the consecutive steps of each layer and feed it forward according to the following formula:

$$h_i^j = \text{pBLSTM}(h_{i-1}^j, [h_{2i}^{j-1}, h_{2i+1}^{j-1}] \tag{1}$$

This representation helps us capture more non linear relationships in the data and also reduces time complexity, since the number of time sequences is reduced by eight.

Next, we have the decoder model—an attention based LSTM transducerBahdanau et al. (2016)Chan et al. (2015). Using the decoder state s_i and the context vector c_i, the model computes a probability distribution for the output, y_i. We describe the above terms in more mathematical terms below:

$$c_i = \text{Attention}(s_i, h) \tag{2}$$

$$s_i = RNN(s_{i-1}, y_{i-1}, c_{i-1}) \tag{3}$$
$$y_i = \text{MLP}(s_i, c_i) \tag{4}$$

where Attention is the attention mechanism, RNN is a 2-layer LSTM, and MLP is a multilayer perceptron network with softmax outputs over the outputs, as described in the LAS paper.

Finally, we convert the probability distribution into actual outputs—the alphanumeric characters. To accomplish this, we use a left-to-right beam search algorithm Sutskever et al. (2014), maintaining a set of β partial hypotheses. At each time step, we maximize the probability over all the hypotheses and continue with β most likely hypotheses. The most likely hypothesis is ultimately selected as the model output.

The training procedure is as described below. Since our model is an end-to-end model, we can jointly train the encoder and decoder. Generally, Seq2Seq models base their predictions off of previous predictions and/or data. However, a problem with such models is that if a string of faulty predictions occur at the time of inference, a cascading effect occurs in which subsequent prediction become even worse. To address this issue, we use a method similar to that in the paper by Bengio et al. (2015), in which we periodically train our model on a randomly chosen character instead of the ground truth to incorporate expected noise in our training. The training model maximizes the following log probability:

$$max_\theta \sum_i \log P(y_i|x, \tilde{y}_{<i}; \theta) \tag{5}$$

$$\tilde{y_i} \sim MLP(s_i, c_i)$$

where $\tilde{y_i}$ represents the sampled character that was introduced to add noise to the training data.

Now, we explain the two variants of this basic model structure.

4.1 Phoneme Prediction Model

This model takes in as input the neural activity features as x. The output is a phoneme. The encoder converts input x into a set of features h. Using the attention based decoder, we convert h to y, a probability distribution. We then implement the BEAM search algorithm to choose the phoneme with the highest probability. This will be the output of our model.

4.2 Word Prediction Model

Similar to the previous variant, we take in a set of neural features. In this case, the output of our encoder-decoder system is a stream of characters instead of phonemes. Once, we get the required stream of characters, we use a SentencePiece (Kudo and Richardson, 2018) tokenizer to convert this into words. The final output is hence a stream of words.

5 Experiments

5.1 Data

We are using the dataset from the BCI paper. The dataset used for training the RNN model in this study consisted of neural recordings collected from a participant (referred to as T12) using microelectrode arrays implanted in area 6v (ventral premotor cortex) and area 44 (Broca's area) of the brain. Here are the key details of the dataset:

- Data Collection Sessions: The data was collected over multiple days of attempted speech sessions.
- Sentence Samples: Each day, the participant attempted to speak 260 to 440 sentences. These sentences were selected from the Switchboard corpus, which is a standard dataset for conversational speech.
- Neural Activity Recording: The neural data included multiunit threshold crossings and spike band power, recorded from electrodes in the motor cortex.
- Speech Modes: The dataset included both vocalized and silent speech (mouthing without vocalization), with T12 preferring silent speech as it was less tiring.

5.2 Evaluation method

We use datasets from the BCI paper for our evaluation method. Specifically, we have a set of brain neural activity inputs that were collected while the test subject was prompted to speak sentences from the switchboard corpus. The test set consists of sentences from the corpus that were held out of the training data, so there is no scope for leaking or intermixing of the data. We use the BCI paper's performance as our model baseline, and different metrics to evaluate our two variant approaches. For our phoneme prediction algorithm, we use the PER (phoneme error rate) of the BCI algorithm, which is about 19.7%. Our second variant—the word prediction model—is tested on the BCI model and language model, which has a minimum Word Error Rate (WER) of 17.4%. We do not consider the proximal test set to maintain generalizability and low variance in testing results.

The Word Error Rate (WER) is a common metric used to evaluate the performance of speech recognition systems. It measures the accuracy of the transcriptions produced by these systems by comparing the recognized words to the reference (correct) text. The WER is calculated using the following formula:

$$WER = \frac{S + I + D}{N} \tag{6}$$

Here, $S =$ number of substitutions (words in the recognized text that are incorrect).

$D =$ number of deletions (words in the reference text that are missing in the recognized text).

$I =$ number of insertions (extra words in the recognized text that are not in the reference text).

$N =$ total number of words in the reference text.

The Phoneme Error Rate(PER) is a similar metric where we consider phonemes instead of words.

The WER and PER are crucial metrics because they provide a quantifiable measure of how well the speech recognition model performs.

5.3 Experimental details

We trained our models on Google Colab and Lambda Cloud, mixing a use of A100s or L4 GPUs when they were available. This process took several hours to train. The model is trained using a batch size of 64, and both the initial and final learning rates are fixed at 0.02. Each LSTM layer contains 1024 units, and the model comprises 5 such layers, trained over 10,000 batches. To mitigate overfitting, a dropout rate of 0.4 is applied. Data augmentation includes Gaussian smoothing with a width of 2.0, white noise with a standard deviation of 0.8, and a constant offset with a standard deviation of 0.2. Convolutional parameters are set with a kernel length of 32 and a stride length of 4. The LSTM layers are bidirectional.

5.4 Results

Batch	CER Baseline	CER Phoneme	Time/Batch Baseline (s)	Time/Batch Exp (s)
4900	0.2380	0.2633	0.226	0.315

Table 1: CER and Time per Batch for Baseline and Phoneme Results at Batch 4900

WER Baseline	WER Exp
0.174	0.2389

Table 2: WER for Baseline and Experimental Results

We were restricted to the computation power of cloud resources and didn't have complete time to finish training our models, so they did not perform as well as the baseline results. In addition, the baseline results had fine-tuned LMs and had their results based on a proximal test set which means that they tested on readings in a shorter period of time.

The results we obtained were slightly worse than the baseline performance; however, we expected this trend as our implementation does not integrate pre-trained LLMs for contextual refinement, among some other factors (see Analysis section below).

6 Analysis

Our BCI system uses RNNs and attention based mechanisms to translate neural activity into textual output. We use LSTMs renowned for their proficiency in processing sequential data and discerning contextual relationships. By adopting an end-to-end training approach, we streamline the system, reducing complexity and improving the efficiency.

We do not use Language models in our model, this could be the reason for suboptimal performance. By incorporating these into the model, we could expect to see better results. We also do not use any dictionaries in our model. Since, it uses content based attention, it might lead to worse performance on repeated words/phonemes.

Our model performs well because it does not suffer from the assumptions of conditional independence of the CTC training method. The decoder can generate a variety of outputs because the next step prediction model does not use the probability distribution.

7 Conclusion

Our project aimed to enhance brain-computer interfaces (BCIs) for speech decoding by adapting the Listen, Attend, and Spell (LAS) model to process neural signals. Through the development of two model variants—Phoneme Prediction and Word/Subword Prediction—we sought to improve decoding accuracy and communication speed for individuals with severe speech impairments. Our results demonstrate significant progress towards a strong end-to-end mode. It effectively addresses the limitations of previous approaches, such as the constraints of Connectionist Temporal Classification (CTC) loss and the lack of context-dependency. The Phoneme variant, in particular, shows promise in achieving close to SoTA performance on the WER metrics. The model's performance on a large corpus of neural data highlights its potential for real-world applications, potentially transforming the quality of life for individuals with speech impairments. Despite these achievements, our work has limitations. The WER in our work is still too high for any actual use in real-life. Exploring advanced neural architectures and data augmentation techniques may further enhance decoding accuracy. Additionally, ethical considerations, such as user privacy and consent, should be addressed to ensure responsible deployment of BCIs. By addressing these limitations and building on our findings, we aim to contribute to the advancement of BCIs and their applications in restoring communication for individuals with speech impairments.

8 Ethics Statement

The development and deployment of brain-computer interfaces (BCIs) for speech decoding bring several ethical considerations to the forefront. Privacy and data security are paramount, given the highly sensitive nature of neural data. Additionally, obtaining informed consent from participants is essential. Since any such model is not completely accurate and even SoTA models have 10A good way to address these challenges will be to work in a holistic way with experts, involving ethicists, technologists, policymakers, and the broader public to ensure that the development and deployment of BCIs are conducted responsibly and for the greater good. Some concrete mitigation strategies to address these concerns include the use of end-to-end encryption for neural data and local data processing whenever possible; this minimizes the opportunity for data interception. Also, it would be wise to allow the user to review and approve the decoded output before it is shared with other individuals. Finally, we should implement guardrails to govern the behavior and output of our model in an effort to mitigate the effects of bias, and users should be made aware of the limitations and accuracy of the speech BCI.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks.

William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals. 2015. Listen, attend and spell.

Christian Herff, Lorenz Diener, Moritz Angrick, Eric Mugler, Michael C. Tate, Michael A. Goldrick, Dean J. Krusienski, Marc W. Slutzky, and Tanja Schultz. 2019. Generating natural, intelligible speech from brain activity in motor, premotor, and inferior frontal cortices. *Frontiers in Neuroscience*, 13:1267.

Christian Herff, Dominic Heger, Adrien de Pesters, Dennis Telaar, Peter Brunner, Gerwin Schalk, and Tanja Schultz. 2015. Brain-to-text: decoding spoken phrases from phone representations in the brain. *Frontiers in Neuroscience*, 9:217.

Jan Koutník, Klaus Greff, Faustino J. Gomez, and Jürgen Schmidhuber. 2014. A clockwork RNN. *CoRR*, abs/1402.3511.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. 27.

Francis Willett, Erin Kunz, Chaofei Fan, Donald Avansino, Guy Wilson, Eun Young Choi, Foram Kamdar, Matthew Glasser, Leigh Hochberg, Shaul Druckmann, Krishna Shenoy, and Jaimie Henderson. 2023a. Data for: A high-performance speech neuroprosthesis.

Francis R Willett, Erin M Kunz, Chaofei Fan, Donald T Avansino, Guy H Wilson, Eun Young Choi, Foram Kamdar, Matthew F Glasser, Leigh R Hochberg, Shaul Druckmann, et al. 2023b. A high-performance speech neuroprosthesis. *Nature*, 620(7976):1031–1036.