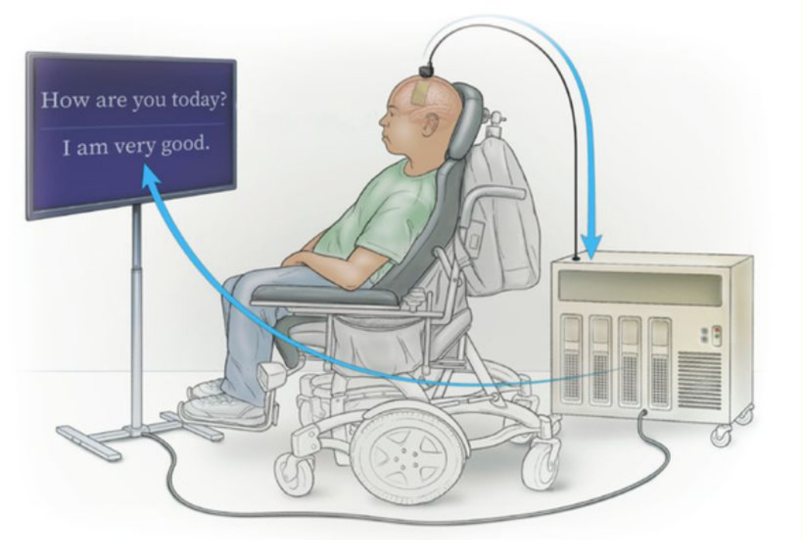
Deep Learning for Brain-Computer Interfaces

Akira Yoshiyama & Lena Podina

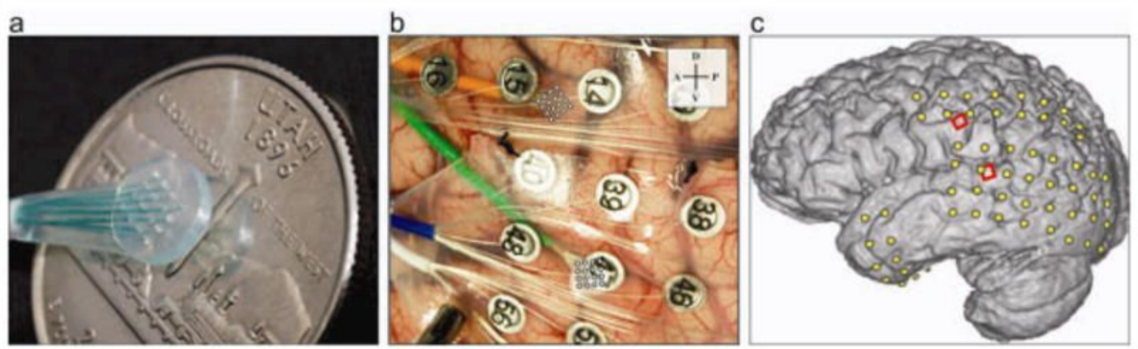
Introduction and Problem Statement

Is it possible to decode neural signals from during attempted speech, and translate it into words, in real time, with a high accuracy and a large vocabulary, using deep neural networks?



Electrocorticography (ECoG)

- Electrocorticography (ECoG) presents an alternative to EEG
  - Implanted electrodes instead of surface of head
- Benefits of ECoG are two-fold:
  (1) Higher spatial and temporal resolution
  (2) Less susceptible to contamination from muscle movements and eye blinks



Past Works

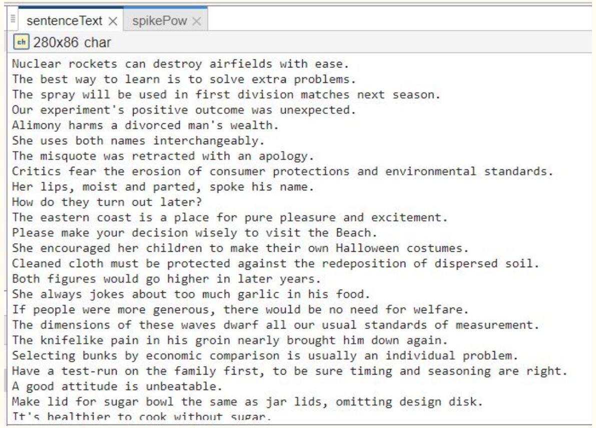
Citation	Method	Performance
Kellis et. al. (2010)	1. PCA to extract features from neural signals 2. Classified words in a predetermined vocabulary	Roughly $85 \pm 13\%$ average accuracy over 45 two-word combinations at best
Pei et. al. (2011)	1. Ranked phonetics using Maximum Relevance and Minimum Redundancy 2. Naive Bayes classifier for final prediction	Avg. classification accuracy for both vowels and consonants ~40%

Past Works

Citation	Method	Performance
Mugler et. al. (2014) **First to propose phonemes**	1. Statistical analysis to identify and rank features 2. Linear Discriminant Analysis (LDA)	Phoneme classification accuracy up to 36% for all, 63% for a single phoneme
Moses et. al. (2021) **Shift to neural networks**	1. Stacked LSTM for speech detection 2. Word classification using two GRUs	Word error rate of 25.6% for a 50 word vocabulary

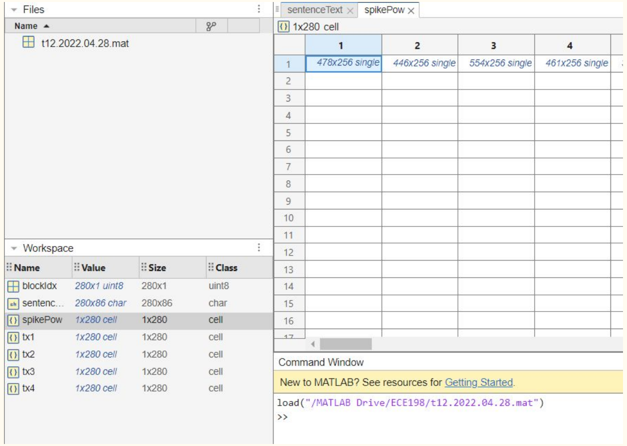
State-of-the-Art Model | Willett et. al. (2023)

- Willett et. al. (2023) created a pipeline to decode ECoG data, achieving record-low error rates and record-high speed
- There are 10,850 sentences in the published dataset
- Dataset consists of spoken sentences and associated neuronal spike power from a 125,000 word vocabulary
- **First successful demonstration of large-vocabulary decoding!**

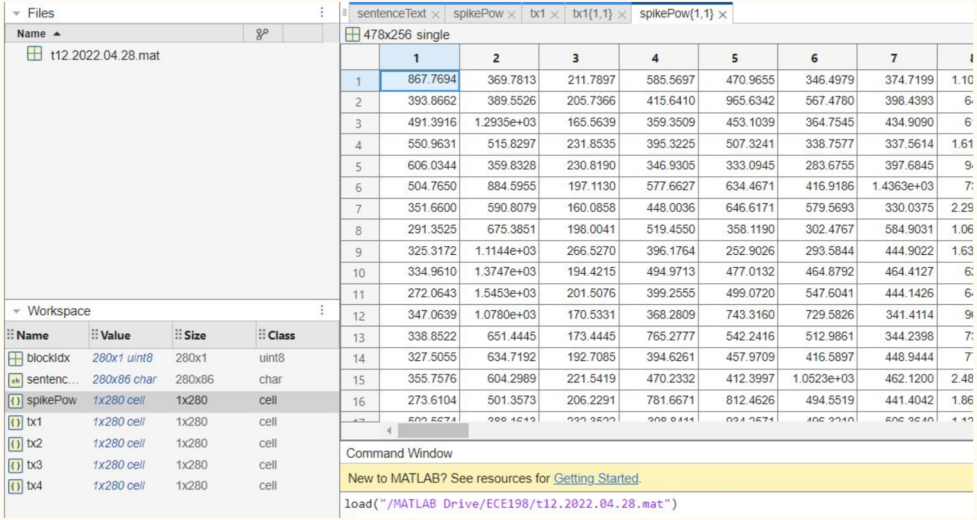


< 280 Sentences

280 TxF Matrices >



Matrix for one sentence (binned spike band power) >



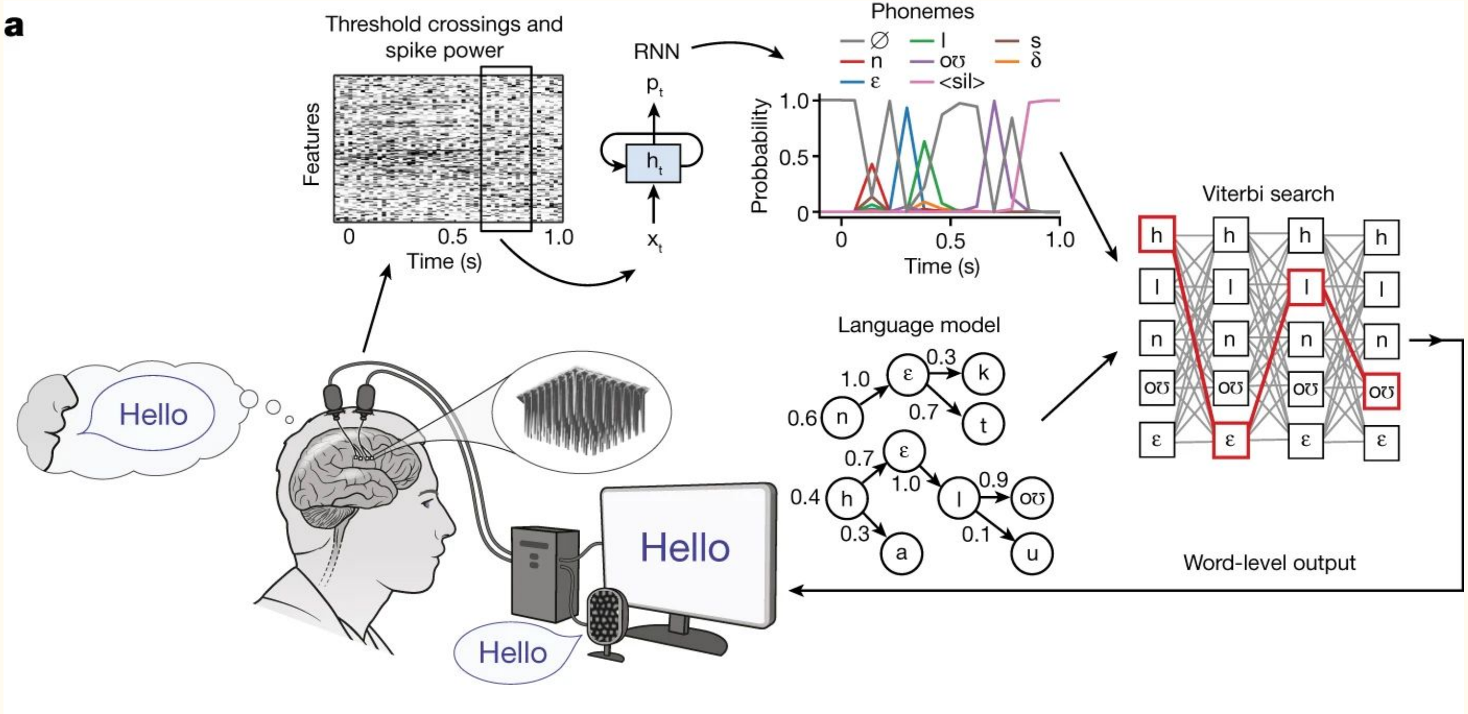
```
                                          ^
                                          |
                                          |
                                      Superior

              Area 44 Superior                                  Area 6v Superior
              192 193 208 216 160 165 178 185     062 051 043 035 094 087 079 078
              194 195 209 217 162 167 180 184     060 053 041 033 095 086 077 076
              196 197 211 218 164 170 177 189     063 054 047 044 093 084 075 074
              198 199 210 219 166 174 173 187     058 055 048 040 092 085 073 072
              200 201 213 220 168 176 183 186     059 045 046 038 091 082 071 070
              202 203 212 221 172 175 182 191     061 049 042 036 090 083 069 068
              204 205 214 223 161 169 181 188     056 052 039 034 089 081 067 066
              206 207 215 222 163 171 179 190     057 050 037 032 088 080 065 064
<-- Anterior                                                                                 Posterior -->
              Area 44 Inferior                                  Area 6v Inferior
              129 144 150 158 224 232 239 255     125 126 112 103 031 028 011 008
              128 142 152 145 226 233 242 241     123 124 110 102 029 026 009 005
              130 135 148 149 225 234 244 243     121 122 109 101 027 019 018 004
              131 138 141 151 227 235 246 245     119 120 108 100 025 015 012 006
              134 140 143 153 228 236 248 247     117 118 107 099 023 013 010 003
              132 146 147 155 229 237 250 249     115 116 106 097 021 020 007 002
              133 137 154 157 230 238 252 251     113 114 105 098 017 024 014 000
              136 139 156 159 231 240 254 253     127 111 104 096 030 022 016 001

                                        Inferior
                                          |
                                          |
                                          â^"
```

Map of placement of Sensor Channels

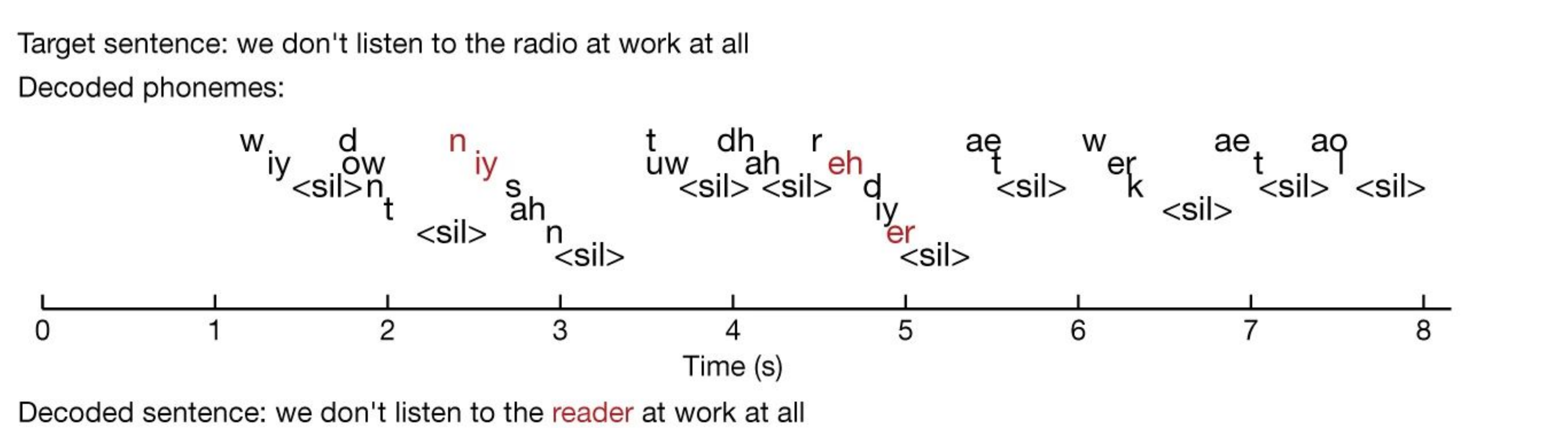
Method | Willett et. al. (2023)



Method | Willett et. al. (2023)

1. Trained a recurrent neural network (RNN) decoder to emit, at each 80 ms time step, the probability of each phoneme being spoken at that time
   a. Used a 5 layer, stacked gated recurrent unit RNN
   b. RNN runs at 4-bin frequency (20 ms bins) outputting a phoneme probability vector every 80 ms
2. These probabilities were then combined with a language model to infer **the most probable underlying sequence of words**, given both the phoneme probabilities and the statistics of the English language
   a. Phoneme errors are often corrected by the language model
   b. Language model translates the sequence of CTC labels into candidate sentences

Method | Willett et. al. (2023)



Example Sentence

Results | Willett et. al. (2023)

- Achieved 9.1% word error rate on 50-word vocabulary
  - 2.7 times fewer errors than previous work
- Achieved 23.8% word error rate for the 125,000-word vocabulary
  - First successful demonstration of large-vocabulary decoding
- Patient spoke at an average pace of 62 words per minute

An End-To-End Alternative

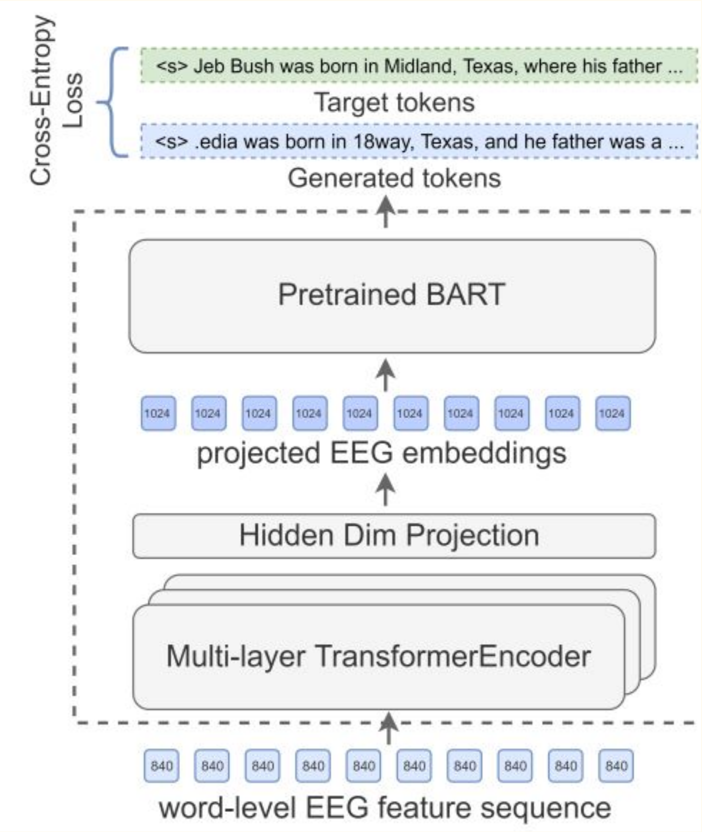
- Common approaches in past works used a chain of models to decode neural activity, with one or more intermediary steps

  What if you have one, end-to-end model? Would that increase accuracy?

- An end-to-end transformer approach might work
  - No work yet applying transformers to ECoG data
  - A paper on EEG-To-Text was recently published, applying transformers to decode EEG signals to attempted speech

An End-To-End Alternative

- Transformers have reached good performance on automated speech recognition
- Wang & Ji (2021) used a pre-trained BART to decode EEG into text
    - Inspired by Speech Recognition (Hinton et al. 2012)
- We are currently attempting to create an Encoder for an ECoG dataset



Thank you!

Q&A Session