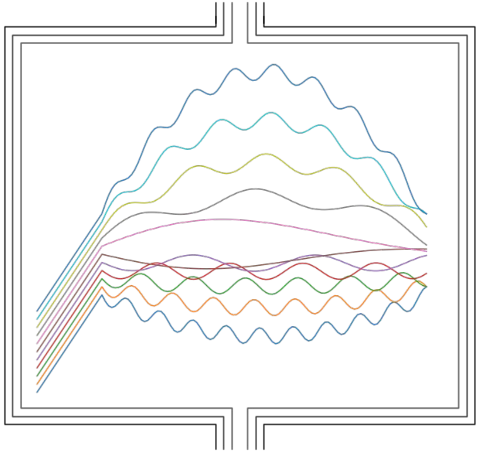
CortexType: Typing with your mind



Neureality Hackathon (March 2024)

Team Electric Sheep:

Hussain Ather, shussainather@gmail.com
Ruilin (Joanna) Qiao, rq2184@columbia.edu
Kateryna (Kate) Shapovalenko, kshapova@andrew.cmu

Table of contents

Meet our team - Electric Sheep!

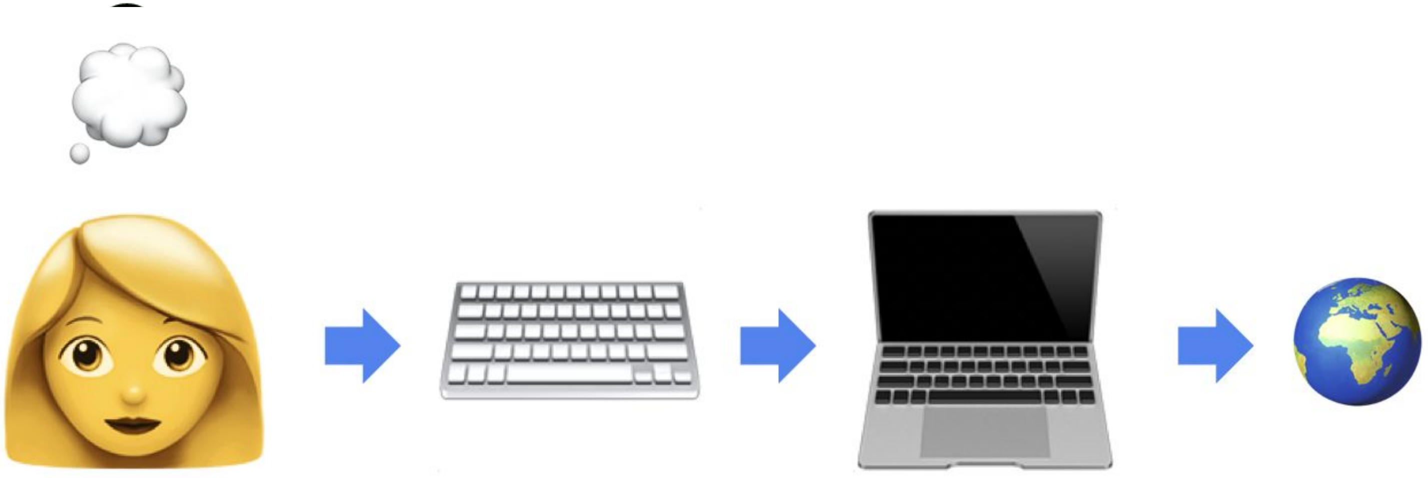
Our teammates (left to right):

@Joanna - biomedical engineering grad at Columbia → responsible for BCI software setup and running tests

@Kate - data science grad at Carnegie Mellon University → responsible for modelling

@Hussain - phd at University of Toronto → participated in the final testing of the algorithm

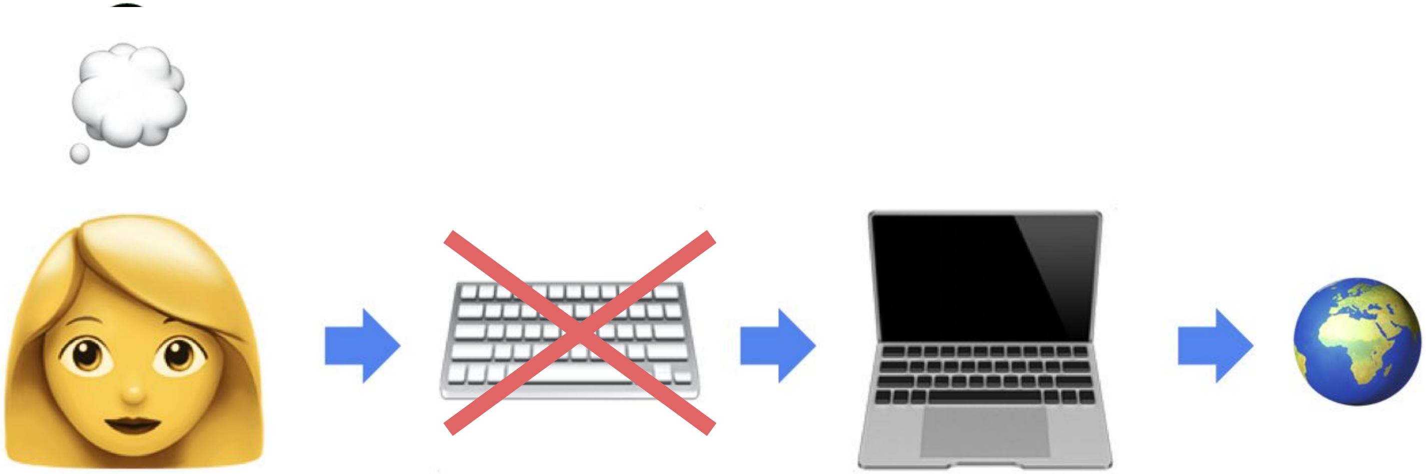
Present communication



Typing is hard, slow, and boring.

But more importantly, it’s not accessible for many people.

What if...



What if there is no physical barrier between your mind and digital interface?

What if you could type just with your mind?

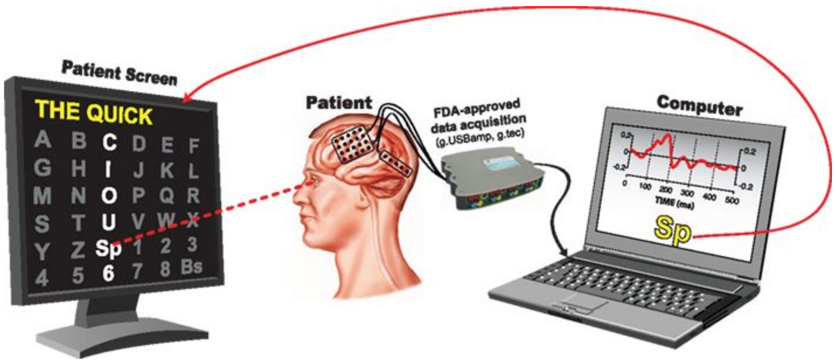
Current tools - P300 BCI Speller

Pros:

- Relatively high accuracy in detecting the focused character.

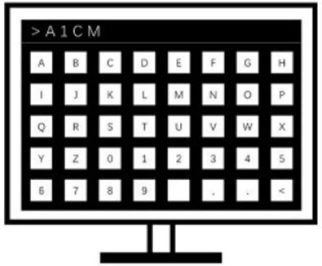
Cons:

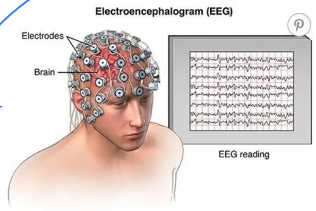
- Require extensive training and calibration for each user.
- Slow typing speed.
- Sensitive to external disturbances.



Our goal - Make the P300 BCI Speller more accurate and fast

Speller visual interface





Stimulus data pre-processing

EEG data preprocessing

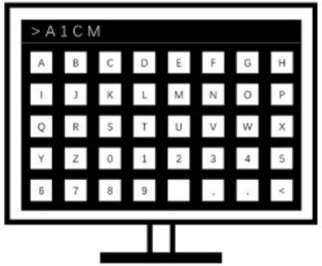
Model training (learn how to associate EEG data and visual stimuli)

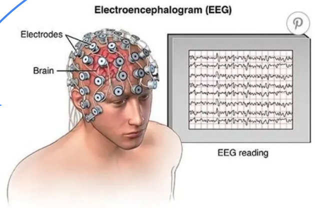
Getting predictions (target character)

Evaluating the results

Improving the model - classifier

Speller visual interface





Stimulus data pre-processing

EEG data preprocessing

Model training

Getting predictions (target character)

Evaluating the results

Improving the model - classifier

Stimulus data

Things we've tried:

Option 1 (default) - Logistic regression

Option 2 - Pre-trained EEGNet

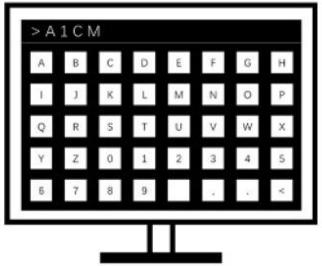
Model training

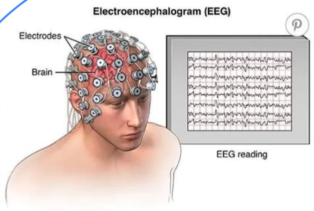
Getting predictions
(target character)

Evaluating the results

Improving the model - getting predictions

Speller visual interface





Stimulus data pre-processing

EEG data preprocessing

Model training

Getting predictions

Evaluating the results

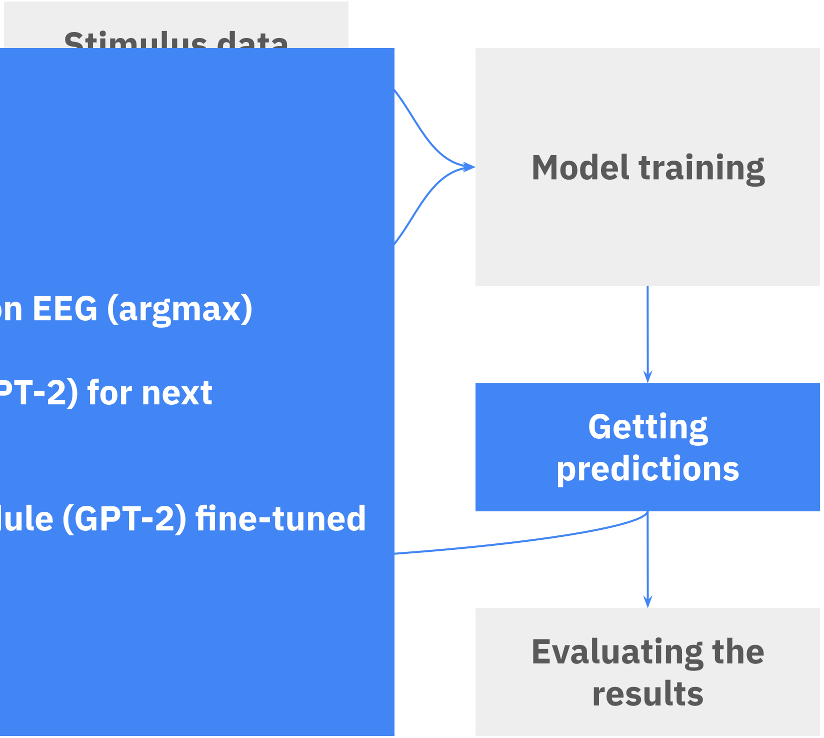
Improving the model - getting predictions

Things we’ve tried:

Option 1 (default) - Simple greedy decoding based on EEG (argmax)

Option 2 - Supplemented with 2 LLMs (BERT AND GPT-2) for next character prediction

Option 3 - Supplemented Option 2 with an LLM-module (GPT-2) fine-tuned on news about neuroscience-related hackathons



Improving the model - getting predictions

Stimulus data

Option 3 example (a mix of LLMs + finetuning):

The subject is trying to spell “N-E-U-R-E-A-L-I-T-Y”

The user already typed in “N-E-U-R-” and currently tries to add “-A”

```
EEG Predicted: A with probability 0.363
Bert Predicted: I with probability 0.009
GPT-2 Predicted: O with probability 0.168
GPT-2 Fine-tuned Predicted: O with probability 0.193
Final Prediction: A
Accumulated Text: NEURA
```

Model training

Getting predictions

Evaluating the results

Training & testing - set up

- **EEG device**: g.tec Unicorn Hybrid Black Headset
- **Human subject**: our teammate Hussain (thanks, Hussain!)
- **Training**: multiple approaches (repeat trials of single or multiple letters)
- **Testing**: spell out the word N-E-U-R-E-A-L-I-T-Y

Training & testing - demo

LINK TO THE DEMO

Evaluation

- **Accuracy**
- **Speed**

Next steps

- Experiment with other EEG signal preprocessing techniques.
- Complete testing and adaptation of models developed during the hackathon the using the EEG equipment.
- Consider introducing additional subject-specific layer to account for variability between users.

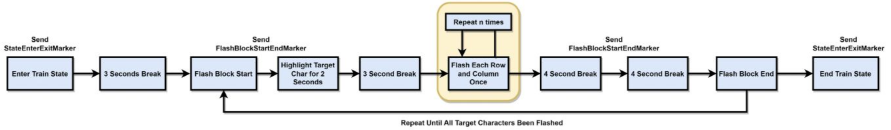
Thanks, organizers! We had fun and learned a lot!

References

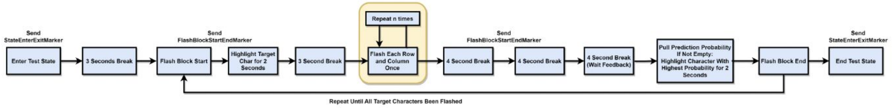
- PhysioLabXR-Community: https://github.com/PhysioLabXR/PhysioLabXR-Community/tree/master
- Neureality Hackathon: https://neureality-cu.github.io/Neureality/hackathon.html
- Pre-trained motor-imagery models: https://neurotechlab.socsci.ru.nl/resources/pretrained_imagery_models/

Annex

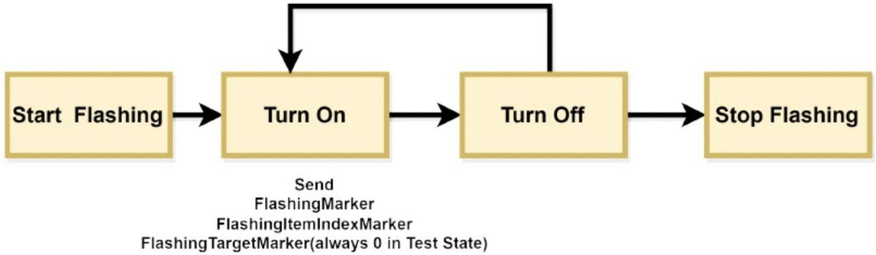
Train State



Test State



Flash Block



Source: https://physiolabxrdocs.readthedocs.io/en/latest/PhysioLabXRP300SpellerDemo.html