Internship Report - MSc Mod4NeuCog (M2)

Fingerprints of structured sequences of events in EEG brain data

Written by *Guilherme Sola dos Santos*

Supervised by *Antonio Galves*

From March 07th to August 26th

São Paulo

August 2022

Summary

1 - Introduction 2

1.1 - Internship activities 4

2 - Materials and Methods 5

2.1 - The Little Words Game 5

2.2 - Experimental Protocol 6

2.3 - Data Analysis 8

3 - Results 14

4 - Discussion 18

5 - Conclusion 19

References 20

Appendix A 22

Appendix B 23

1 - Introduction

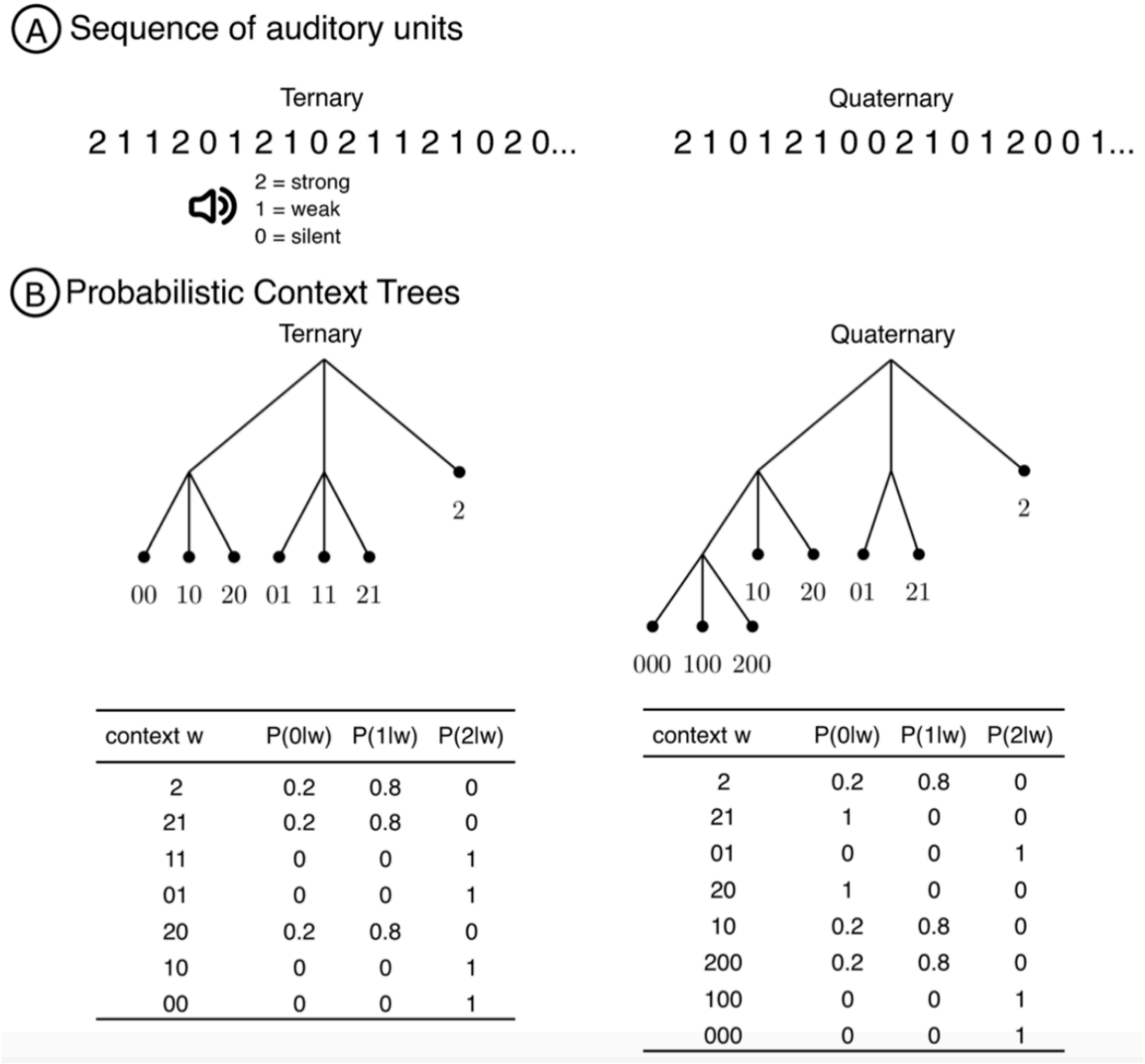
The goal of this project is to use a novel theoretical framework recently introduced by the NeuroMat project to develop a Brain-Computer Interface application that allows people to communicate through brain signals, translating them into intended words. We want to know: is it possible to predict a person's future writing intentions while typing a text just by analyzing brain activity data?

The brain's ability to detect statistical regularities from sequences of stimuli was first conjectured by von Helmholtz [1]. Further known as *statistical learning*, evidences from this ability were observed in experiments with different types of stimuli [2-4]. One way to work with this conjecture is to identify signatures of statistical regularities in brain activity associated with the sequence of stimuli presented [5]. The ability to detect these regularities is crucial to the task of classifying and predicting the situation to which an individual is exposed, playing an important role in human decision making [6].

In parallel to that, while observing sequences of data in the real world, Rissanen introduced a universal data compression model named *context tree* [7]. It follows the logic that new symbols in sequences are dependent only of a relevant past and are chosen probabilistically. This relevant past of symbols containing all the information needed to choose the new symbols are called *contexts*. Every context has a transition probability associated with it and the set of all relevant contexts can be organized into a tree representation [Figure 1]. Context trees can be used to compress various types of sequences and can be employed to model linguistic phenomena [8].

Inspired by von Helmholtz [1] and Rissanen [7] contributions, Duarte et al. proposed a new statistical approach to analyze the electroencephalography (EEG) data generated by a participant exposed to a sequence of auditory stimuli. This new approach made it possible to retrieve a context tree from the statistical regularities of brain signals by modeling the relation between the sequence presented and EEG segments [9].

With a framework for retrieving a probabilistic context tree model from brain data available, the question raised was whether this approach would be able to retrieve a context tree of a sequence of stimuli compatible with the context tree previously used to generate that specific sequence. Fortunately, Hernández et al. showed that context trees generating sequence of auditory stimuli can effectively be extracted from EEG data [10] by employing the statistical model proposed in Duarte et al. [9].



context w	P(0|w)	P(1|w)	P(2|w)
2	0.2	0.8	0
21	0.2	0.8	0
11	0	0	1
01	0	0	1
20	0.2	0.8	0
10	0	0	1
00	0	0	1

context w	P(0|w)	P(1|w)	P(2|w)
2	0.2	0.8	0
21	1	0	0
01	0	0	1
20	1	0	0
10	0.2	0.8	0
200	0.2	0.8	0
100	0	0	1
000	0	0	1

Figure 1 - Probabilistic Context Trees models of sequences (Hernández et al., 2021) [10]. In (A), two different sequences of auditory stimuli. In (B), the context trees together with their transition probabilities. With this scheme, it is possible to extract all the information necessary to construct a new sequence generated by the information compressed in the context trees. Given a context, one can infer the next unit to appear in the sequence based on the transition probabilities.

Now, if this information extraction is possible, can it be used in a Brain-Computer Interface approach? Brain-Computer Interface (BCI) is the use of brain signals to control devices [11]. A classic BCI approach is to perform text typing tasks, such as the P300 Speller (a BCI system that uses P300 waves to allow the participants to control a virtual keyboard displayed on a computer screen) [12].

The combination of the novel statistical model selection approach to the identification of the laws governing segments of EEG recorded while a participant is exposed to a structured and random sequence of stimuli [9] and the idea to use this theoretical framework to develop a BCI application that allows people to communicate through brain signals, translating them into intended words, brought us to the following question:

Is it possible to predict a person's future writing intentions while typing a text just by analyzing brain activity data?

1.1 - Internship activities

As an intern at the Center for Innovation and Diffusion in Neuromathematics (NeuroMat - FAPESP), I first spent three weeks in Rio de Janeiro working at the Deolindo Couto Institute of Neurology (INDC/UFRJ), led by Claudia Vargas. There I had the opportunity to run some EEG experiments using the Goalkeeper Game [13], a video game developed by NeuroMat in which the participant has to control a goalkeeper and defend sequential penalty kicks. The kicker decides in which direction to kick according to a sequence generated from a context tree.

With the Goalkeeper Game, I was introduced to the new theoretical framework [9] developed by the NeuroMat team. To understand this framework, I had to study some theory around the topics: sequences generated by Stochastic Chains, information compression by Context Trees [7], dimensionality reduction using the Projective Method [14], cumulative distributions analysis and the Kolmogorov-Smirnov test. Once comfortable with these topics, we began to discuss possible ways to explore this new approach to develop an innovative BCI application.

During this internship I had the opportunity to develop a complete experimental research project. Starting by defining the question to be pursued and an initial strategy to reach the answer, going through the discussion of different experimental protocols, contemplating advantages and disadvantages between them. With the experimental protocol chosen, I had to develop and implement it from scratch, including programming the port connections between the experiment algorithm and the EEG system. After everything was ready, I had to collect data by running EEG experiments with some participants, analyze the data and discuss the results with the NeuroMat research team. For this to be possible, I had to invest a lot of time programming algorithms to

implement and to make the statistical framework [9] compatible with a BCI approach (all scripts are available on github).

In addition, I also had the opportunity to participate of several discussions around other theoretical topics (Wasserstein distance, Markov Chains, Gaussian Mixture Models) and meetings for discussing articles in preparation for submission.

All the challenges faced during this internship in NeuroMat definitely helped me to improve my skills as a researcher and as a professional.

2 - Materials and Methods

Is it possible to predict a person's future writing intentions while typing a text just by analyzing brain activity data?

The first task in order to answer this question was to define an experimental protocol. We were inspired by an experiment introduced by Shannon [15] where he successively presents letters of a word and asks the participant to try to guess the word in question after presenting each new letter. In this study, instead of successively presenting letters to the participant, we chose to leave the participant free to type words given a certain set of letters.

To simplify the problem, we decided leave just a few letters available for the participant and explored the possibility of predicting the intended word before the participant concludes the typing action. If we succeed in doing it, the BCI application could be some kind of word auto-completer supported only by the information acquired from the EEG.

2.1 - The Little Words Game

For the experiment session, a game was specifically developed for this research. It is called ‘The Little Words Game’ [Figure 2A]. The goal of the game is to find the maximum number of words as possible in every round using only the available letters displayed on the screen. In each round, every new word found will be fixed on the screen. At the end of each round all words will be counted, the record will be updated and the words will be reseted to a new round to start.

There are a few rules presented to the participant before the game starts:

- It is only allowed to use each letter once per word;
- Only words formed by 3 or more letters are considered valid words;
- After typing all the letters necessary to complete the intended word, press ‘Enter' to confirm it;
- Repeated words in the same round will not be counted;
- In each round, it is allowed to type any word found in previous rounds.

In case any of these rules are broken, a warning message appears on the screen informing the reason why the intended word was not accepted as a valid word.

2.2 - Experimental protocol

For the purpose of this research, we only let 4 letters available (‘A’, ‘I’, ‘M’, ‘R’). We will denote the set of all possible letters and the character that represents the space between words as $U = \{$'A', 'I', 'M', ‘R’, ‘_’$\}$.

Following the rules of the game, it is possible to write 10 words with those letters in Brazilian Portuguese language ('IMA' - *magnet*, ‘IRMA’ - *sister*, 'IRA' - *rage*, ‘MAR' - *sea*, ‘MIA’ / ‘MIAR’ - *meow*, ‘MIRA' - *aim*, ‘RIA’ - *laugh*, ‘RIM’ - *kidney*, ‘RIMA' - *rime*) according to the list contemplating all Brazilian Portuguese words prepared by the Department of Computer Science of the Institute of Mathematics and Statistics of the University of São Paulo (IME/USP) [16]. Any word not included in this list was not considered a valid word.

Until now, all 3 native Brazilian Portuguese speakers participants played 180 rounds that lasted for 20 seconds each (total of 60 minutes playing the game). They seated comfortably around 85cm away from the screen and used a Cedrus RB-840 Response Pad controller where each button was linked to a letter (‘A' - *green*, ‘I' - *blue*, ‘M' - *red*, ‘R’ - *yellow*, 2x ‘Enter' - *transparent*). Both hands where used during the experiment. The middle and index fingers from the left hand where used for the letters ‘A' and ‘I’, respectively, and the index and middle fingers from the right hand for the letters ‘M' and ‘R’, respectively. Since there were two buttons linked to the ‘Enter' input, the thumb finger from any of the hands could be used to press a button that activate this function. The other two white buttons of the controller had no function [Figure 2B].

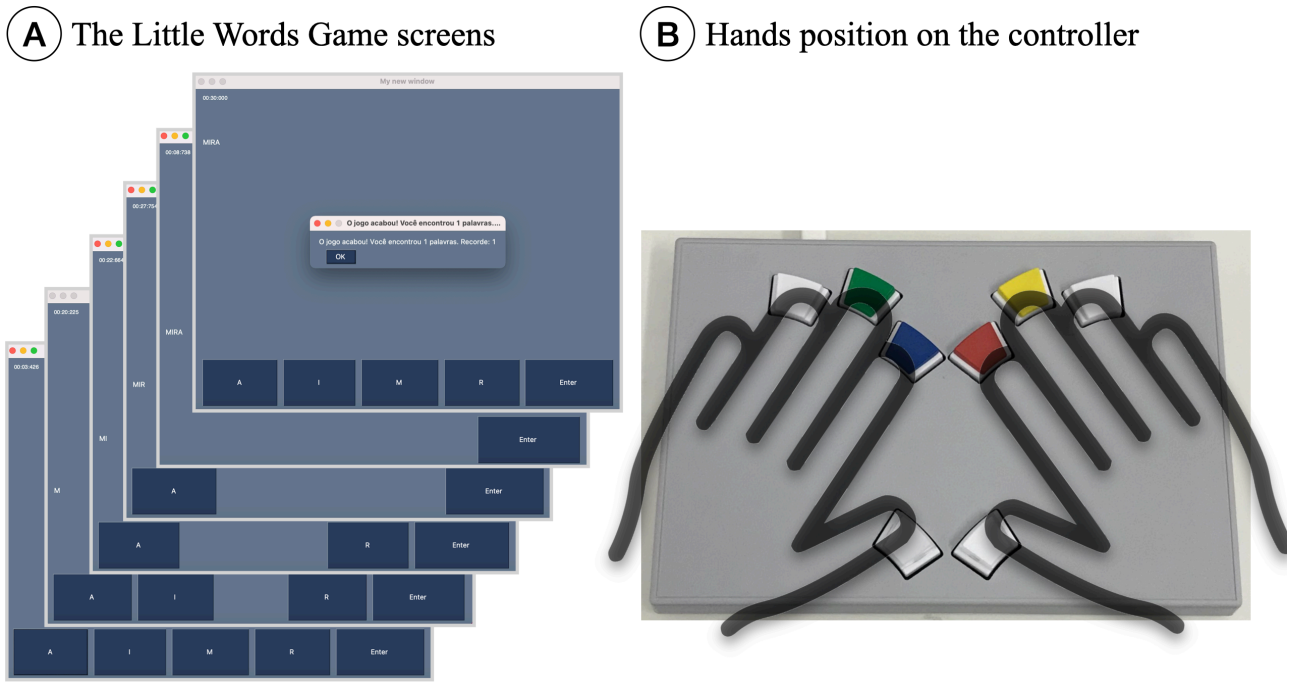


Figure 2 - The Little Words Game. In (A) it is possible to see the game evolution while the participant typed a word. On the bottom of the screen, all letters and an 'Enter' key were provided. As soon as the participant hit a key it disappears. When the 'Enter' button was hit, every button became available again. Once the game was over or if any rule was broken, a message appears. In (B) the hands positions are represented on top of the controller.

Before starting to play the game, the participant had the 31+2 electrodes EEG cap (Brain Products GmbH actiCAP) put on. Let E be the set with all the 31 electrodes:

E = {Fp1, Fp2, F7, F3, Fz, F4, F8, FC5, FC1, FC2, FC6, FT9, FT10, C3, C4, T7, T8, CP5, CP1, CP2, CP6, TP9, TP10, P7, P3, Pz, P4, P8, O1, Oz, O2}.

The electrodes were grounded at Fpz, referenced at Cz and connected to a Brain Products GmbH ActiCHamp amplifier (1000-Hz sampling rate).

Conductive gel was applied to each electrode until an impedance level around 10-20k Ohms were reached. The data was acquired using the Brain Vision Recorder software. The recorded signal was filtered with a band-pass filter of 1–30 Hz and a baseline by mean correction was later performed on segmented data.

Markings on the EEG data were made for all participant actions and game responses. They were later used to segment the data and to retrieve all the valid words found by the participant and the moment in which the actions associated with every valid word happened (average of 5ms and standard deviation of 3ms of delay).

2.3 - Data Analysis

Since the goal is to find out if it is possible to predict a person's future writing intentions by figuring out the word being written before the participant concludes the typing action, the problematic becomes a classification task. We will try to predict the next letter $u \in U$ that will be typed by the participant p given a prefix.

Let's suppose that the participant p found the following list of 9 words ['IMA', 'IRMA', 'IRA', 'MAR', 'MIA', 'MIRA', 'RIA', 'RIM', 'RIMA']. Now, an example, let's focus on the letter 'R'. The letter 'R' appears in the following words ['IRMA', 'IRA', 'MAR', 'MIRA', 'RIA', 'RIM', 'RIMA'].

Considering this set of words, the appearance of letter R leads to 4 different letters $u \in U$, which are: 'M' ['IRMA'], 'A' ['IRA', 'MIRA'], '_' ['MAR'] and 'I' ['RIA', 'RIM', 'RIMA']. This means that only observing the letter 'R' does not give us enough information to define what letter is coming next. In this scenario, we adapted the logic introduced by Rissanen [7] to simplify our classification task.

By observing the 'R' connected to one more past unit we can find the following structures: 'IR' ['IRMA', 'IRA', 'MIRA'], 'AR' ['MAR'], '_R' ['RIA', 'RIM', 'RIMA']. Note that, by doing it, we are able to determine that if a sequential unit 'AR' appears, the next letter $u \in U$ to be typed will be a '_' with probability 1. At the same time, if a sequential unit '_R' appears, the next letter $u \in U$ to be typed will be a 'I' also with probability 1. Every sequential unit that leads to any $u \in U$ with probability 1 will be considered a prefix.

For the sequential unit 'IR' there are still two possible next letters $u \in U$ to be typed, the letters 'M' and 'A'. Here, we can try to look further for one more unit in the past. This will result in the following structures: '_IR' ['IRMA', 'IRA'] and 'MIR' [MIRA]. By doing it we found one more prefix 'MIR', but the sequential unit '_IR' still does not give us enough information to infer with probability 1 the next letter $u \in U$.

Note that we can not apply the same strategy again since there is no more past units before '_IR'. In this case, '_IR' will be considered a prefix that may leads to two different letters $u \in U$. The transition probabilities associated with this prefix will be defined by the relative frequency of '_IR' leading to the letter 'A' and '_IR' leading to the letter 'M' observed in the participant p performance while playing the game.

In this example, we could see that considering the list of words [‘IMA’, ‘IRMA’, ’IRA', ‘MAR', ‘MIA’, ‘MIRA', ‘RIA’, ‘RIM’, ‘RIMA’], we can infer which letter $u \in U$ will be typed after the letter ‘R' observing four different prefixes [‘AR', ‘_R’, ‘MIR’, ‘_IR’]. In the case of the prefix [‘_IR’], we will need to solve a classification problem in order to determine the next letter $u \in U$ that will be typed.

Note that, in the case of the structure ‘MI' [‘MIA’, ‘MIRA'], following the strategy of observe 'MI' with one more past unit, we get the structure ‘_MI’ [‘MIA’, ‘MIRA’]. In this case, where the new sequential unit starts with ‘_’ and there is no gain for the classification task, we ignore the sequential unit ‘_MI’ and the sequential unit ‘MI' is considered a prefix. We will refer to this situation as *inefficient past unit addition*.

The logic to define the set of prefixes given a list of words is expressed in Algorithm 1 as pseudocode.

Algorithm 1 - Pseudocode describing the process to define a set of prefixes given a list of words

Input: List of words and set of letters U
Output: Set of prefixes

```
Creation of an empty sequential_unit_list
for u ∈ U do
  Include u in the sequential_unit_list
  for sequential_unit ∈ sequential_unit_list do
    right_letters_list ← set of all letters u ∈ U that appears after the sequential_unit in the list of words
    if length(right_letters_list) > 1 do
      if sequential_unit starts with '_' do
        if inefficient past unit addition is True do
          Removal of '_'
          Inclusion of the sequential_unit to the set of prefixes (associated with transition probabilities)
        else do
          Inclusion of the sequential_unit to the set of prefixes (associated with transition probabilities)
        end if
      else do
        Include all sequential_units with one more past unit in the sequential_unit_list
      end if
    else do
      Inclusion of the sequential_unit to the set of prefixes (transition probability = 1)
    end if
```

20: **end for**
21: **end for**
22: **return** Set of prefixes

From now on, let denote prefix as ω and let τ be the set of all prefixes that leads to at least two different letters $u \in U$. It is important to note that the words found by each participant p throughout the game rounds may be different, which means that the set τ generated for the participant p depends on the individualized data. This means that the number of prefixes $\omega \in \tau$ and the transition probabilities associated with them are unique per participant [Figure 3A].

Knowing all the prefixes $\omega \in \tau$, it is possible to search for them in the EGG data. We observed the times when these prefixes appeared in the EEG data to define the size of a window used to segment the data. The window size (w) is defined by the greatest time distance between the last action related to a word and the first appearance of a prefix of the next word [Figure 3B]. The intention of doing this is to access the greatest amount of information before every prefix without invading any signal segment related to a typing action of another word. Since the participants do not type at the same speed, the window size is not necessarily the same between the participants.

To analyze the problem by a classification point of view, the data of electrode e is randomly divided into two datasets: Training and Test datasets (where 80% of the data was designated for training a predictive model and 20% for testing the accuracy of the model in predicting the next letter u that will be typed). Let $\omega \in \tau$ be a fixed prefix and the appearance of the prefix ω followed by a letter $u \in U$ be ωu. We denote by $Y^{\omega u,e} = (Y_1^{\omega u,e}, \cdots, Y_N^{\omega u,e})$ the list of all the EEG segments assigned to the ωu condition in the training sample collect from a fixed electrode $e \in E$. Note that each $Y_n^{\omega u,e}$ depends on the choice of w. To simplify the notation, if there is no danger of confusion, in general we will omit mentioning the electrode e in the notation [Figure 3C].

To work with a classification task problem we used the following analysis inspired by the approach introduced in Duarte et al. [9]. This approach applies the Projective Method introduced in Cuesta-Albertos et al. [14] that transforms a function into a real number by projecting that function in a random direction determined by a Brownian Bridge. Let us call B the realization of the Brownian Bridge.

A Generation of the set of prefixes

U = Letters available + '_'

```
['A', 'I', 'M', 'R', '_']
```

Words typed by the participant p

```
({'RIMA': 150,
  'IMA': 169,
  'MIA': 136,
  'MAR': 155,
  'RIM': 143,
  'MIRA': 137,
  'IRA': 154,
  'RIA': 107,
  'IRMA': 95})
```

Algorithm 1

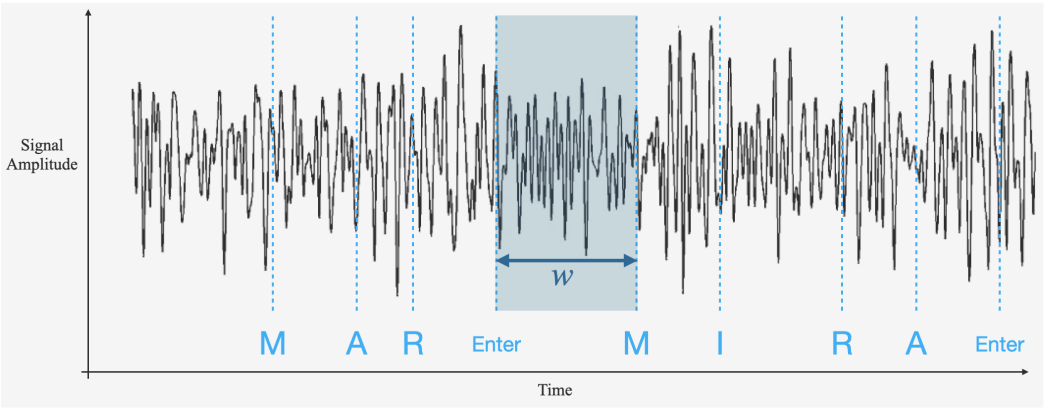
Set of all prefixes associated with transition probabilities

```
{'IA': {'freq': 243, '_': 1.0},
 'RA': {'freq': 291, '_': 1.0},
 '_I': {'freq': 418, 'M': 0.404, 'R': 0.596},
 'MI': {'freq': 273, 'A': 0.498, 'R': 0.502},
 'RI': {'freq': 400, 'A': 0.268, 'M': 0.733},
 '_M': {'freq': 428, 'A': 0.362, 'I': 0.638},
 'RM': {'freq': 95, 'A': 1.0},
 '_R': {'freq': 400, 'I': 1.0},
 'AR': {'freq': 155, '_': 1.0},
 '_MA': {'freq': 155, 'R': 1.0},
 'IMA': {'freq': 319, '_': 1.0},
 'RMA': {'freq': 95, '_': 1.0},
 '_IM': {'freq': 169, 'A': 1.0},
 'RIM': {'freq': 293, '_': 0.488, 'A': 0.512},
 '_IR': {'freq': 249, 'A': 0.618, 'M': 0.382},
 'MIR': {'freq': 137, 'A': 1.0}}
```

- Prefix
- How many times this prefix was observed in the participant p performance
- Next letter $u \in U$ to appear and transition probabilities associated to it

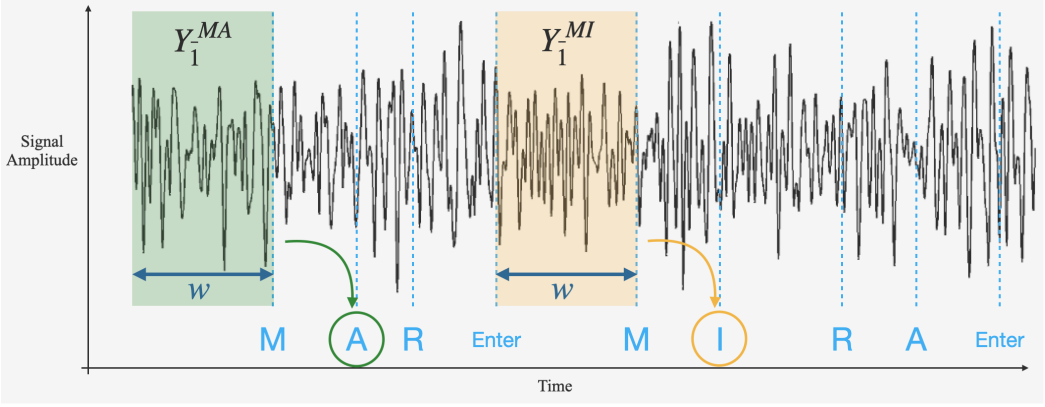
B Window Size w definition

Segment of EEG data from training dataset



C EEG segments assigned to the ωu condition

Segment of EEG data from training dataset



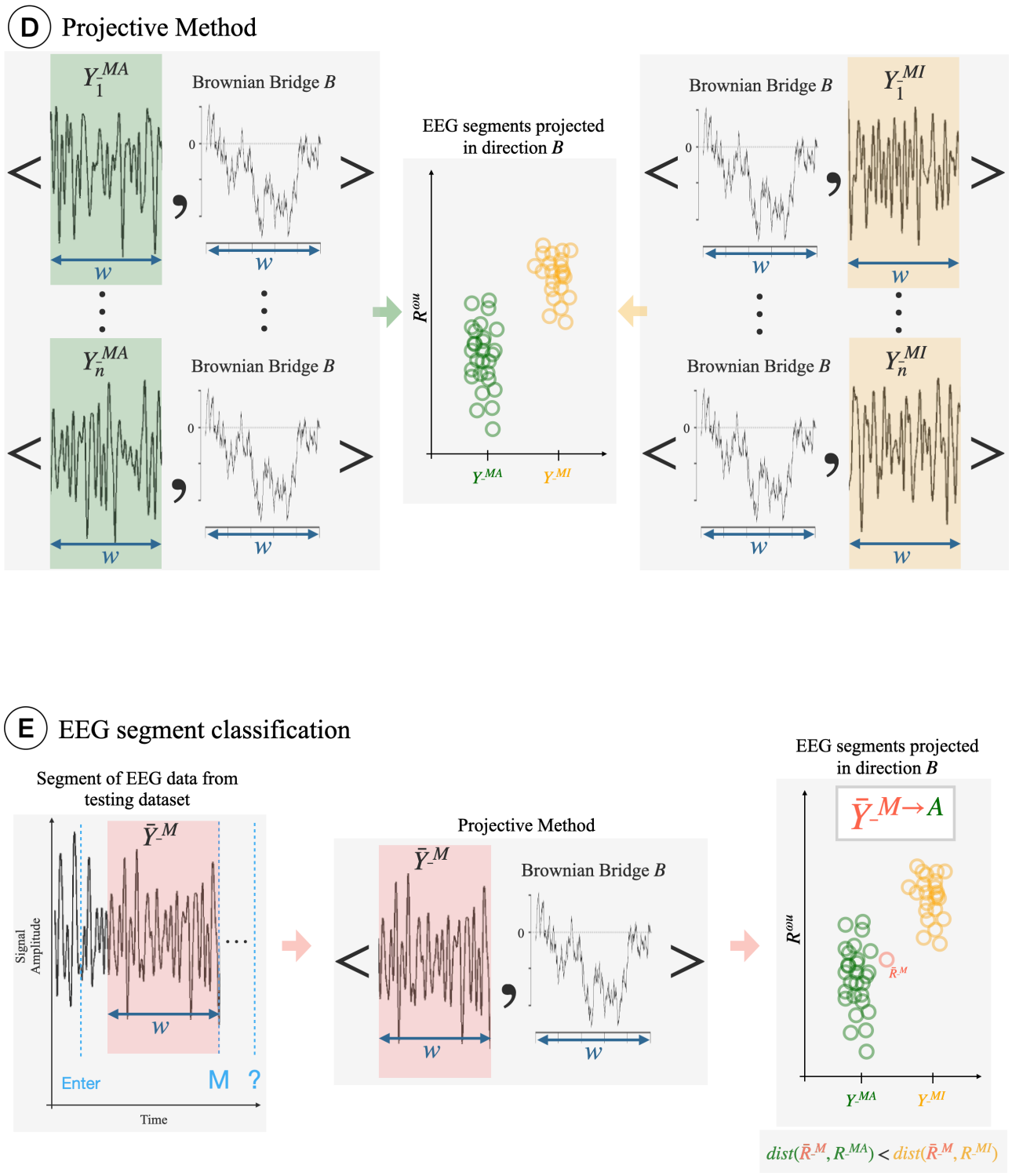


Figure 3 - Data Analysis Scheme. In (A) a set of prefixes generated using the logic described in Algorithm 1. It is the combination of a group of available letters *U* and the words typed by the participant *p* while playing the game. In (B) a representation on the window size *w* definition. Followed by (C) where the prefix '_M' is found in some EEG segments from the training dataset. The first one leads to the letter 'A' and the second one to the letter 'I'. In (D), considering the group of all EEG segments of the prefix '_M' encountered in the training dataset, a scheme to illustrate the projective method and the set of segments projected in a a realization of a Brownian Bridge *B*. Finally, in (E), a representation of the classification process of a new segment of the prefix '_M' encountered in the testing dataset.

To apply this method in a classification task we first project, for every $u \in U$, the EEG segments $Y^{\omega u}$ of the training dataset in a direction, that is, we take the inner product between a realization B of the Brownian Bridge and each of the EEG segments. By doing it, we acquire the set $R^{\omega u} = (R_1^{\omega u}, \cdots, R_N^{\omega u})$ of real numbers where $R_n^{\omega u} = < Y_n^{\omega u}, B >$ [Figure 3D].

Let $\bar{Y}^{\omega}$ be an EEG segment from the test dataset we want to classify, that is, some EEG segment from some test set associated with ωu, for some $u \in U^{\omega}$, where $U^{\omega} = \{u \in U : |Y^{\omega u}| > 0\}$. Hereafter, we can take the projection of $\bar{Y}^{\omega}$ in the same direction B as the train dataset was projected. We call this real number $\bar{R}^{\omega}$.

Now, we just need to decide to which set from the training sets associated with the prefix ω better describes the segment we want to classify. In order to do it, we calculate the average Euclidian distance between $\bar{R}^{\omega}$ and each training point $R_n^{\omega u}$. This gives us:

$$dist(\bar{R}^{\omega}, R^{\omega u}) = \frac{1}{N} \sum_{i=1}^{N} |\bar{R}^{\omega} - R_i^{\omega u}|$$

We classify the EEG segment $\bar{Y}^{\omega}$ by $\bar{u} = \arg_{u \in U^{\omega}} \min [dist(\bar{R}^{\omega}, R^{\omega u})]$. [Figure 3E]

Finally, for robustness, we repeat this process M times, taking a new realization of the Brownian Bridge for each iteration. We observe which of the letters $\bar{u}$ is the most frequently classified for that EEG segment and give it as the final classification decision. When having all final decisions for every segment, we compare the classification with the original letter u that appeared after that EEG segment to calculate the accuracy of prediction from our model.

In order to have a more complete view of the accuracy scenario, we repeat the entire procedure for all EEG electrodes $e \in E$.

The data analysis procedure description is provided in Algorithm 2 as pseudocode.

Algorithm 2 - Pseudocode describing the data analysis procedure

---

Input: EEG data collected while a participant plays 'The Little Words Game'.

Output: Table of prediction accuracy of the next letter u to be typed given a prefix ω for every EEG electrode $e \in E$.

1: Generation of a unique τ associated with the participant's performance in the game (Algorithm 1)

2: Definition of the specific window size w depending on participant typing speed

3: EEG data segmentation based on every $\omega \in \tau$ and the window size w

4: **for** $e \in E$ **do**
5: Randomly data division into two datasets (80% training data / 20% testing data)
6: **for** $\omega \in \tau$ **do**
7: **for** $m = 1$ to M **do**
8: Determination of a Brownian Bridge B
9: Calculation of $R_n^{\omega u} = < Y_n^{\omega u}, B >$
10: **for** $\bar{Y}^{\omega} \in data_{test}$ **do**
11: Calculation of $\bar{R}^{\omega}$
12: Calculation of $\bar{u} = \arg_{u \in U^{\omega}} \min [dist(\bar{R}^{\omega}, R^{\omega u})]$
13: Comparison u and $\bar{u}$
14: Accuracy metric update
15: **end for**
16: **end for**
17: **end for**
18: **end for**
19: **return** Table of prediction accuracy

3 - Results

Applying our data analysis methodology, the first result obtained was the set of prefixes for each participant. The first participant p_1 found 9 words while playing the game (all except 'MIAR') and the participants p_2 and p_3 found all 10 possible words (for detailed results look Appendix A). As consequence, two different sets of prefixes τ were generated:

$\tau_1 = \{$'_I', 'MI', 'RI', '_M', 'RIM', '_IR'$\}$ and $\tau_{2,3} = \{$'_I', 'MI', 'RI', '_M', 'RIM', '_IR', 'MIA'$\}$.

The second step in the analysis was to define the window size for each participant. For p_1 the window size was $w_1 = 176ms$. Similarly, p_2 window size was $w_2 = 132ms$. The third participant p_3 was a faster typer and had a window size $w3 = 34ms$.

Knowing the windows sizes, it was already possible to run the rest of the procedure in order to classify EEG segments, but one questions still remained: how many M Brownian Bridges realizations are optimal to project the EEG segments in terms of accuracy and efficiency?

To answer that question we run some simulations with the data from participants p_1 and p_2 for the electrode $e =$ 'C3'. We tested the evolution of the accuracy level of classification for scenarios growing by 50 Brownian Bridges realizations until it reaches $M = 1000$. We observed a

stabilization of accuracy levels around 200 Brownian Bridges realizations for several electrodes and decided to proceed our analysis with $M = 200$. [Figure 4].

With the number of Brownian Bridges M defined, we could proceed to the classification step of our analysis. We applied the projective method 200 time for every electrode $e \in E$ and for every prefix $\omega \in \tau_n$ for all participants.

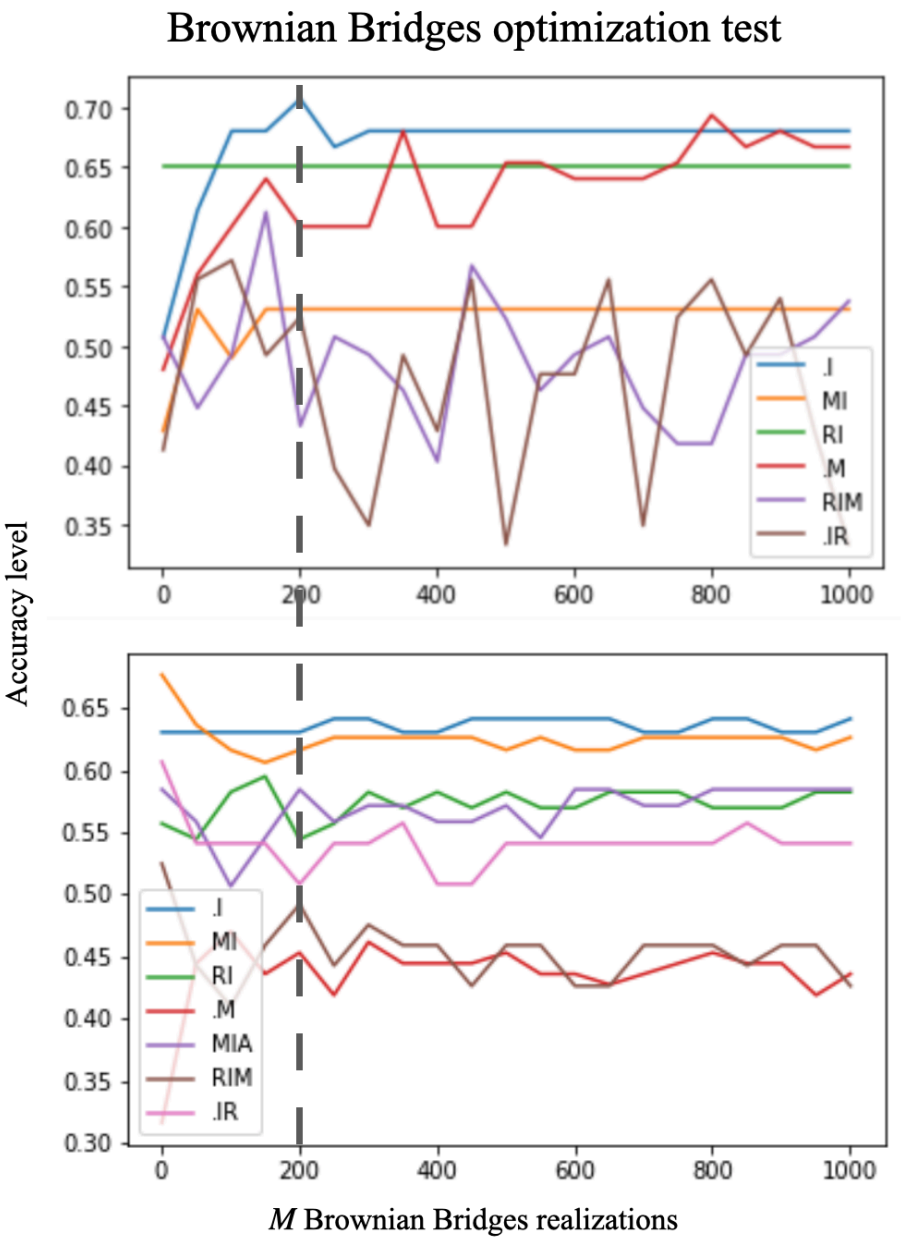


Figure 4 - Brownian Bridge realizations test. When the simulation reaches 200 Brownian Bridges realizations it is possible to observe the accuracy levels stabilizing for most of prefix. Because of this behavior, we chose to work with $M = 200$ for the rest of analysis.

For the participant p_1, the classification accuracy for the prefix ‘RI' stood out. The model reached an average of 72% and a maximum of 80% accuracy in predicting the next letter u to be typed. More than that, all electrodes showed a good performance in the prediction task [Figure 5A].

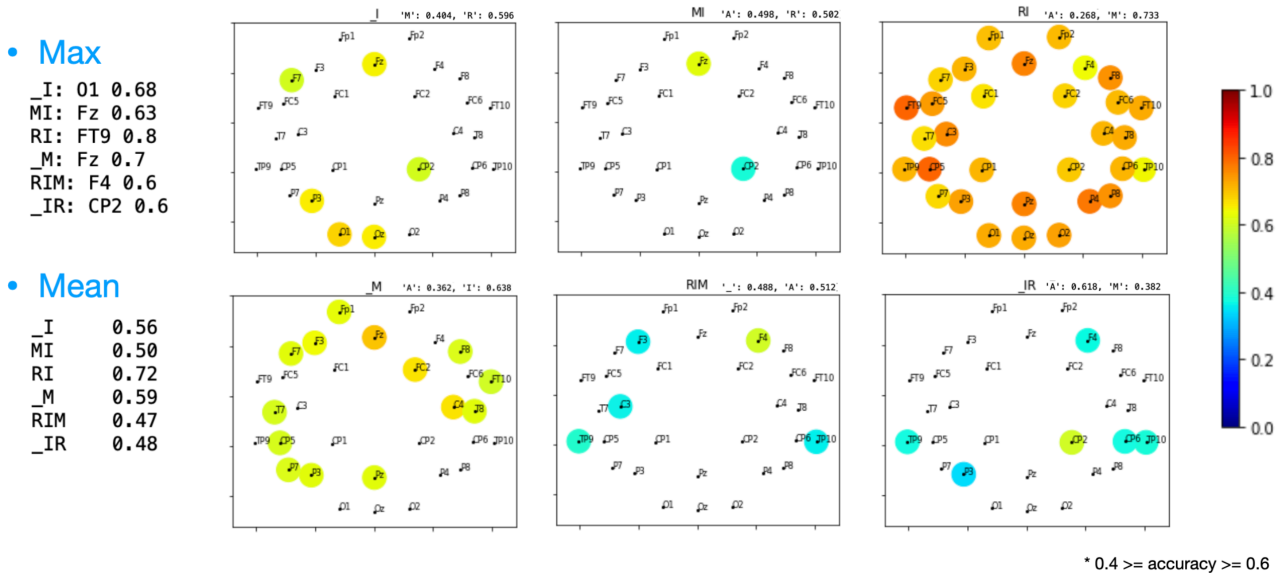
The results observed for the prefix ‘_M’ also revealed an interesting average accuracy level of 59%, with maximum of 70%. The results for the rest of electrodes stayed very close to 50% accuracy in average, with only a few electrodes accuracy level reaching more than 60%.

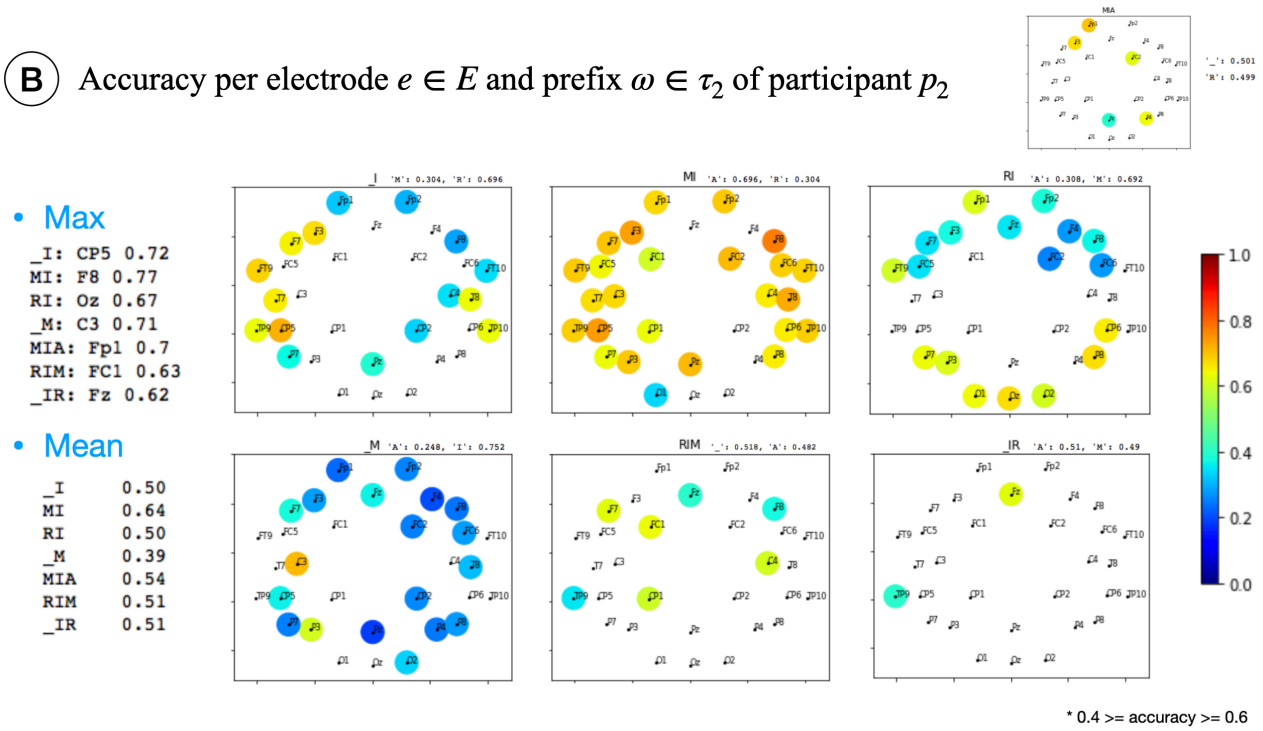
For the participants p_2 and p_3, the best results appeared for the prefix ‘MI’. For both the model reach a maximum accuracy level higher than 70% and average around 60% [Figure 5B/5C].

It is interesting to notice the proximity in the results obtained by the participants p_2 and p_3 in comparison with participant p_1. Remember that participant p_1 have a unique set of prefixes τ_1, while the participants p_2 and p_3 share a same set of prefixes.

Furthermore, the most interesting result observed was the fact that, for all participants, it seems that there is a dependence between the transition probabilities related to next possible letters to be typed and the accuracy levels reached by the model. Note the poorest accuracy rates for prefixes ‘MI’, ‘RIM’ and ‘_IR’ for the participant p_1, and the prefixes ‘MIA', ‘IR' and 'RIM' for the participants p_2 and p_3. All these prefixes share the fact that the transition probabilities associated with the next letters that can be typed are close to an equilibrium of 50% / 50%.

Ⓐ Accuracy per electrode $e \in E$ and prefix $\omega \in \tau_1$ of participant p_1





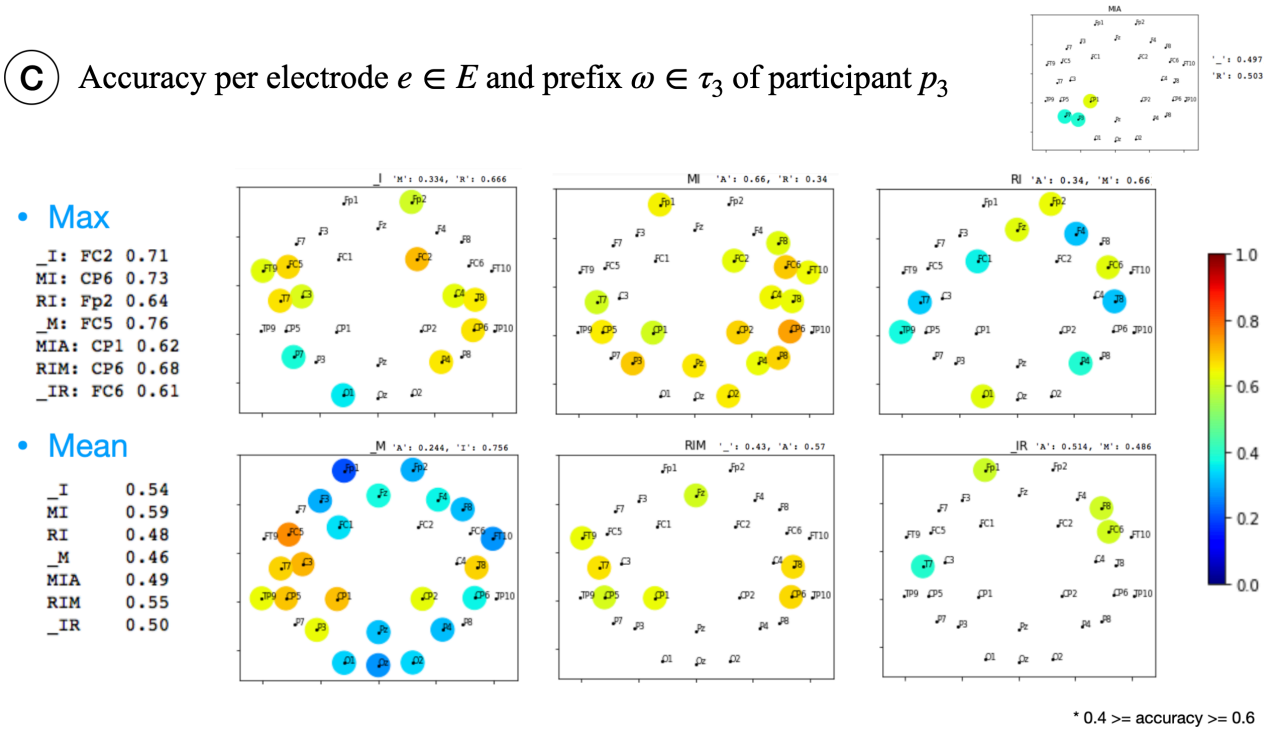


Figure 5 - Accuracy per electrode and prefix for each participant. (A), (B) and (C) represents the results for participant p_1, p_2 and p_3 respectively. On left of the images there is a summary of the maximum and mean classification accuracy reached for every prefix. For the maximum metric, it also shows in which electrode the performance was the best. On top of each graphic, it is referenced the prefix ω and also the possible next letters u to appear, with the transition probabilities associated to it. For the participants p_2 and p_3, the extra prefix compared to the set of prefix for the participant p_1 ('MIA') is positioned on the top right corner. The color of the circles around each electrode represents the accuracy level. Accuracy levels between 0.4 and 0.6 were erased from the graphic for better visualization.

On the other hand, all the prefixes that revealed the best results in term of accuracy have a disproportionate transition probabilities associated with the next letters that can be typed. It is the case of the prefix ‘MI' for the participants p_2 and p_3 (70% / 30% and 66% / 34%, respectively) and the prefixes ‘_M’ and ‘RI' for the participant p_1 (36% / 64% and 27% / 73%). It is also interesting the fact that these prefixes mostly appears in the start of words.

One last thing that is important to note is that the transition probabilities are correlated to the number of possible words to be found in the game that contains the prefixes on it. Takes as example the prefix ‘MI’. For the participant p_1 this prefix appears in only two words [‘MIA’, ‘MIRA’]. If the participant p_1 typed both words the same amount of time during the game, the transition probabilities associated with the next letter being ‘A' and ‘R' would be exactly 50% / 50%. But for the participants p_2 and p_3 the prefix ‘MI' appears in three words [‘MIA’, ‘MIAR', ‘MIRA’]. The scenario where all three words were typed the same amount of time would result in transition probabilities of 66.6% for letter ‘A' and 33.3% for letter ‘R’.

4 - Discussion

Given the results obtained, it is important to point the fact that it looks like it is easier to observe traces of statistical regularities from EEG data when there is an unbalanced equilibrium between the expectations. The bias in the expectation on "what is coming up next" seems to generate a signal that can be detected and classified using the projective method.

Further, the fact that the best results were encountered on prefixes that appear in the beginning of words might reveals that the brain anticipates suffixes instead of just the next letter to be typed. More than that, it seems that the brain adapts pretty well with the restrictions imposed by the game. Otherwise, it would be hard to observe this relation between the unbalanced transitions probabilities and the accuracy power, since there are many other words in the Brazilian Portuguese language containing the prefixes analyzed in this research.

Finally, it is important to mention some limitations of this work. The first one is the low number of participants and, on top of that, all native Brazilians playing a game in Brazilian Portuguese language. Another important observation is that some words (like the word ‘IRMA’) are written with an accent (‘IRMÃ’) that change the sound of the pronunciation of the letter. This might impact the EEG signals and was not taken into account during the analysis.

It is also possible to critique the window size variation between the participants and the set of parameter M. Thinking of that, we run the some tests changing the parameters and we did not find any result that made us believe necessary a review in the decisions taken during the data analysis (look Appendix B). We also experimented using only the 20% best Brownian Bridges realizations in terms of generating less equal distributions (according to the Kolmogorov-Smirnov test).

5 - Conclusion

Is it possible to predict a person's future writing intentions while typing a text just by analyzing brain activity data?

Given the evidences in this research it seems it is possible to predict a person's future writing intentions just by analyzing brain activity data.

The strategy of analyzing segments of EEG data using the projective method have a lot of potential and has to be further studied in order to be better understand. An improvement of the knowledge around the topic and a better application of the technique could lead to new possibilities of Brain-Computer Interfaces.

References

[1] von Helmholtz, H. **Handbuch der physiologischen Optik.** vol. III (Leopold Voss, 1867). Translated by The Optical Society of America in 1924 from the third germand edition, 1910, Treatise on physiological optics.

[2] Frost, R. et al. **Domain generality versus modality specificity: the paradox of statistical learning.** Trends in Cognitive Sciences, v. 19, p. 117–125 (2015).

[3] HuntT, R. H.; Aslin, R. N. **Statistical learning in a serial reaction time task: Access to separable statistical cues by individual learners.** Journal of Experimental Psychology. General, v. 130, p. 658–680 (2001).

[4] Conway, C.M.; Christiansen, M. H. **Sequential learning in non-human primates**. Trends in Cognitive Sciences, v. 5, p. 539–546 (2005).

[5] Dehaene, S. et al. **The Neural Representation of Sequences: From Transition Probabilities to Algebraic Patterns and Linguistic Trees**. Neuron, v. 88, p. 2–19 (2015).

[6] Summerfield, C.; de Lange, F. P. **Expectation in perceptual decision making: Neural and computational mechanisms**. Nat. Rev. Neurosci. 15, 745–756. https://doi.org/10.1038/nrn3838 (2014).

[7] Rissanen, J. **A universal data compression system**. IEEE Trans. Inf. Theory 29, 656–664. https://doi.org/10.1109/TIT.1983.1056741 (1983).

[8] Galves, A.; Galves, C.; García, J. E.; Garcia, N. L.; Leonardi, F. **Context tree selection and linguistic rhythm retrieval from written texts**. Ann. Appl. Stat. 6, 186–209 (2012).

[9] Duarte, A.; Fraiman, R.; Galves, A.; Ost, G.; Vargas, C. D. **Retrieving a context tree from EEG data.** Mathematics7, 427. https://doi.org/10.3390/math7050427 (2019).

[10] Hernández, N.; Duarte, A.; Ost, G.; Fraiman, R.; Galves, A.; Vargas, C. D. **Retrieving the structure of probabilistic sequences of auditory stimuli from EEG data**. Scientific reports, 11(1), 1-15 (2021).

[11] Vidal, J. J. **Toward Direct Brain-Computer Communication.** Annual Review of Biophysics and Bioengineering, vol. 2, pp. 157-180, (1973).

[12] Clerc, M.; Mattout, J.; Maby, E.; Devlaminck, D.; Papadopoulo, T.; et al. **Verbal Communication through Brain Computer Interfaces**. Interspeech - 14th Annual Conference of the International Speech Communication Association - Frédéric Bimbot, Aug 2013, Lyon, France. hal-00842851 (2013)

[13] Stern R. B.; d'Alencar M. S.; Uscapi Y. L.; Gubitosov M. D.; Roque A. C.; Helene A. F.; Piemonte M. E. P. **Goalkeeper game: A new assessment tool for prediction of gait performance under complex condition in people with parkinson's disease**. Frontiers in Aging Neuroscience, 12:50 (2020).

[14] Cuesta-Albertos, J. A.; Fraiman, R.; Ransford, T. **Random projections and goodness-of-fit tests in infinite-dimensional spaces.** Bull. Braz. Math. Soc. New Ser. 37, 477–501. https://doi.org/10.1007/s00574-006-0023-0 (2006).

[15] Shannon, C. E. **Prediction and entropy of printed english.** Bell Systems Technical Journal, 30: 50–64 (1951)

[16] Feofiloff, P. **Lista de todas as palavras do português brasileiro.** (Transl.: *List of all Brazilian Portuguese words*). Department of Computer Science of the Institute of Mathematics and Statistics of the University of São Paulo (IME/USP) - https://www.ime.usp.br/~pf/dicios/ [Updated on 2022-03-22]. Accessed on April 11, 2022

Appendix A

Generation of the set of prefixes for participant p_1

U = Letters available + '_'

```
['A', 'I', 'M', 'R', '_']
```

Words typed by the participant $p1$

```
({'RIMA': 150,
  'IMA': 169,
  'MIA': 136,
  'MAR': 155,
  'RIM': 143,
  'MIRA': 137,
  'IRA': 154,
  'RIA': 107,
  'IRMA': 95})
```

Set of all prefixes associated with transition probabilities

```
{'IA': {'freq': 243, '_': 1.0},
 'RA': {'freq': 291, '_': 1.0},
 '_I': {'freq': 418, 'M': 0.404, 'R': 0.596},
 'MI': {'freq': 273, 'A': 0.498, 'R': 0.502},
 'RI': {'freq': 400, 'A': 0.268, 'M': 0.733},
 '_M': {'freq': 428, 'A': 0.362, 'I': 0.638},
 'RM': {'freq': 95, 'A': 1.0},
 '_R': {'freq': 400, 'I': 1.0},
 'AR': {'freq': 155, '_': 1.0},
 '_MA': {'freq': 155, 'R': 1.0},
 'IMA': {'freq': 319, '_': 1.0},
 'RMA': {'freq': 95, '_': 1.0},
 '_IM': {'freq': 169, 'A': 1.0},
 'RIM': {'freq': 293, '_': 0.488, 'A': 0.512},
 '_IR': {'freq': 249, 'A': 0.618, 'M': 0.382},
 'MIR': {'freq': 137, 'A': 1.0}}
```

- Prefix
- How many times this prefix was observed in the participant p performance
- Next letter $u \in U$ to appear and transition probabilities associated to it

Generation of the set of prefixes for participant p_2

U = Letters available + '_'

```
['A', 'I', 'M', 'R', '_']
```

Words typed by the participant $p2$

```
({'MIRA': 153,
  'MAR': 166,
  'RIM': 142,
  'RIA': 122,
  'MIA': 176,
  'MIAR': 175,
  'IRA': 154,
  'RIMA': 132,
  'IMA': 132,
  'IRMA': 148})
```

Set of all prefixes associated with transition probabilities

```
{'RA': {'freq': 307, '_': 1.0},
 '_I': {'freq': 434, 'M': 0.304, 'R': 0.696},
 'MI': {'freq': 504, 'A': 0.696, 'R': 0.304},
 'RI': {'freq': 396, 'A': 0.308, 'M': 0.692},
 '_M': {'freq': 670, 'A': 0.248, 'I': 0.752},
 'RM': {'freq': 148, 'A': 1.0},
 '_R': {'freq': 396, 'I': 1.0},
 'AR': {'freq': 341, '_': 1.0},
 'MIA': {'freq': 351, '_': 0.501, 'R': 0.499},
 'RIA': {'freq': 122, '_': 1.0},
 '_MA': {'freq': 166, 'R': 1.0},
 'IMA': {'freq': 264, '_': 1.0},
 'RMA': {'freq': 148, '_': 1.0},
 '_IM': {'freq': 132, 'A': 1.0},
 'RIM': {'freq': 274, '_': 0.518, 'A': 0.482},
 '_IR': {'freq': 302, 'A': 0.51, 'M': 0.49},
 'MIR': {'freq': 153, 'A': 1.0}}
```

- Prefix
- How many times this prefix was observed in the participant p performance
- Next letter $u \in U$ to appear and transition probabilities associated to it

Generation of the set of prefixes for participant p_3

U = Letters available + '_'

```
['A', 'I', 'M', 'R', '_']
```

Words typed by the participant $p3$

```
({'MIA': 164,
  'MIRA': 170,
  'RIA': 158,
  'MIAR': 166,
  'RIMA': 175,
  'MAR': 161,
  'IRA': 171,
  'IRMA': 162,
  'IMA': 167,
  'RIM': 132})
```

Set of all prefixes associated with transition probabilities

```
{'RA': {'freq': 341, '_': 1.0},
 '_I': {'freq': 500, 'M': 0.334, 'R': 0.666},
 'MI': {'freq': 500, 'A': 0.66, 'R': 0.34},
 'RI': {'freq': 465, 'A': 0.34, 'M': 0.66},
 '_M': {'freq': 661, 'A': 0.244, 'I': 0.756},
 'RM': {'freq': 162, 'A': 1.0},
 '_R': {'freq': 465, 'I': 1.0},
 'AR': {'freq': 327, '_': 1.0},
 'MIA': {'freq': 330, '_': 0.497, 'R': 0.503},
 'RIA': {'freq': 158, '_': 1.0},
 '_MA': {'freq': 161, 'R': 1.0},
 'IMA': {'freq': 342, '_': 1.0},
 'RMA': {'freq': 162, '_': 1.0},
 '_IM': {'freq': 167, 'A': 1.0},
 'RIM': {'freq': 307, '_': 0.43, 'A': 0.57},
 '_IR': {'freq': 333, 'A': 0.514, 'M': 0.486},
 'MIR': {'freq': 170, 'A': 1.0}}
```

- Prefix
- How many times this prefix was observed in the participant p performance
- Next letter $u \in U$ to appear and transition probabilities associated to it

Appendix B

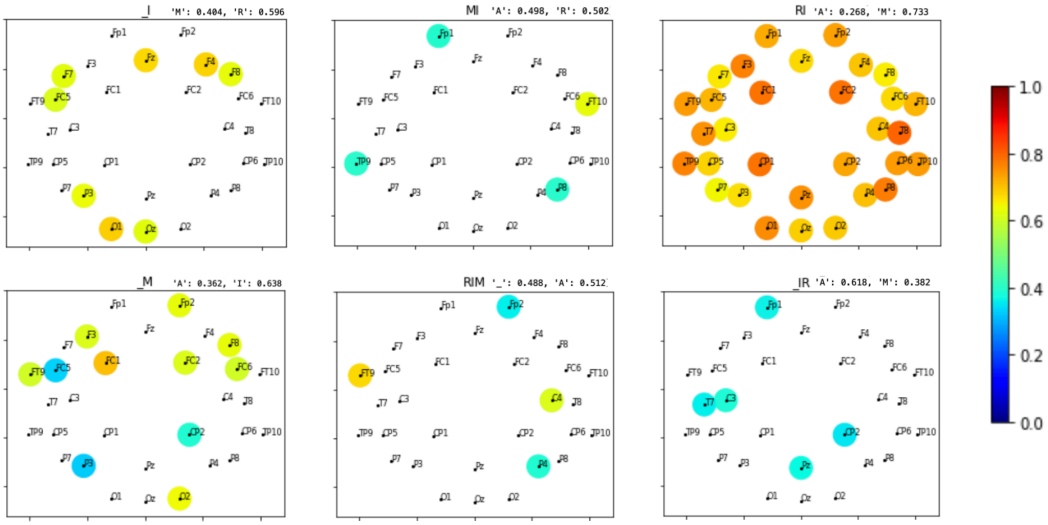
Accuracy per electrode $e \in E$ and prefix $\omega \in \tau_1$ of participant p_1 ($w = 300ms$)

- Max

_I: O1 0.69
MI: FT10 0.63
RI: T8 0.8
_M: FC1 0.71
RIM: FT9 0.68
_IR: CP1 0.59

- Mean

_I	0.55
MI	0.49
RI	0.72
_M	0.54
RIM	0.53
_IR	0.47



* 0.4 >= accuracy >= 0.6

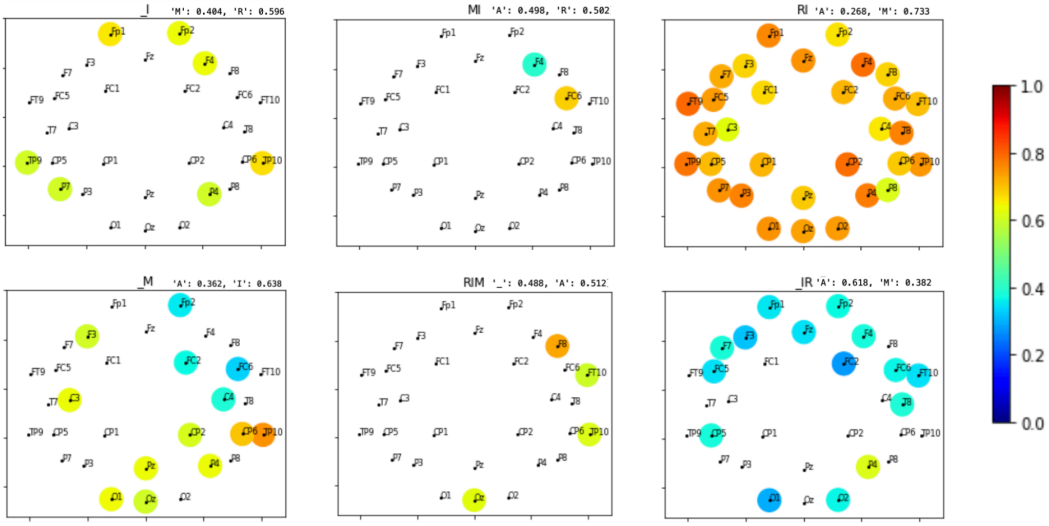
Accuracy per electrode $e \in E$ and prefix $\omega \in \tau_1$ of participant p_1 ($w = 500ms$)

- Max

_I: TP10 0.67
MI: FC6 0.69
RI: FT9 0.79
_M: TP10 0.75
RIM: F8 0.73
_IR: P4 0.62

- Mean

_I	0.56
MI	0.51
RI	0.72
_M	0.52
RIM	0.53
_IR	0.43



* 0.4 >= accuracy >= 0.6

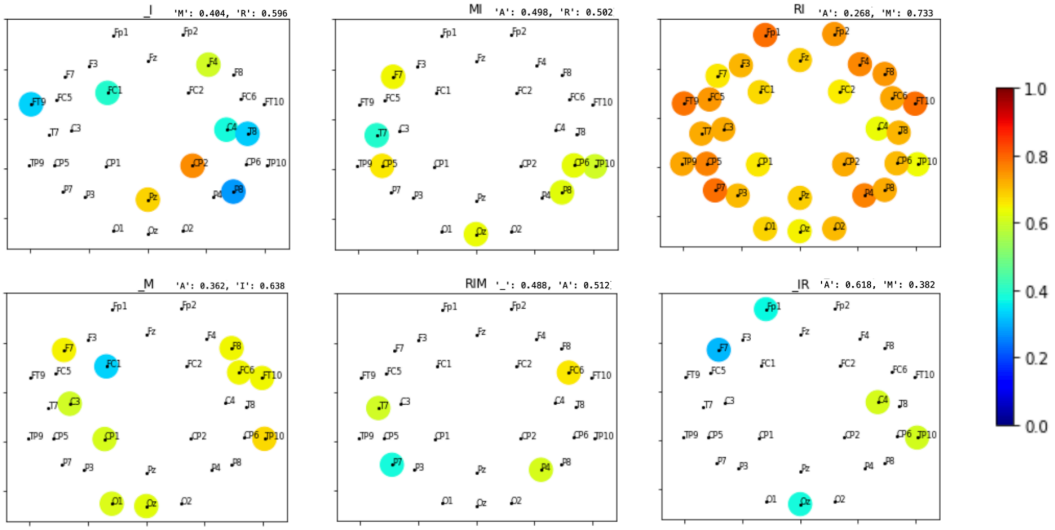
Accuracy per electrode $e \in E$ and prefix $\omega \in \tau_1$ of participant p_1 ($M = 800$)

- Max

```
_I: CP2 0.76
MI: CP5 0.66
RI: Fp1 0.79
_M: TP10 0.68
RIM: FC6 0.66
_IR: C4 0.61
```

- Mean

_I	0.51
MI	0.52
RI	0.72
_M	0.55
RIM	0.53
_IR	0.49



* 0.4 >= accuracy >= 0.6

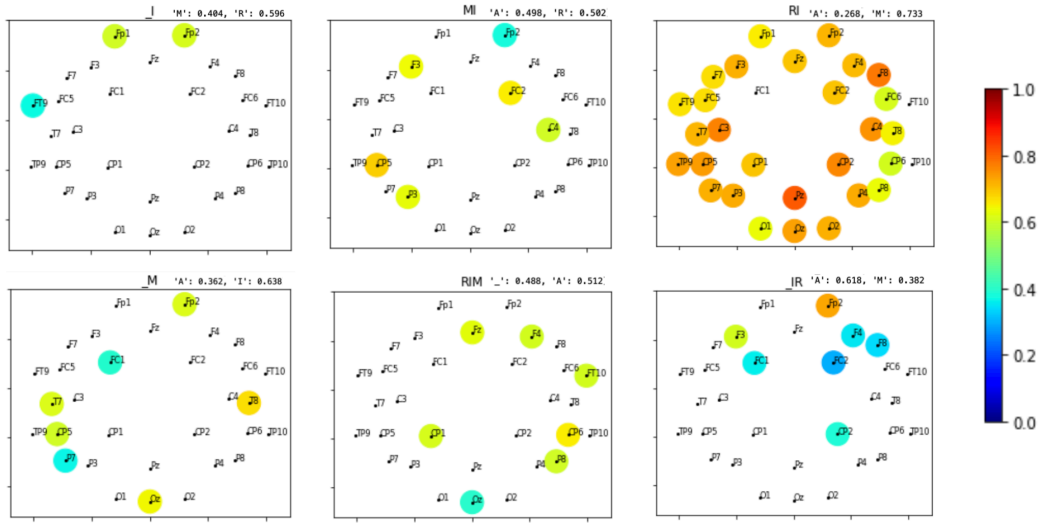
Accuracy per electrode $e \in E$ and prefix $\omega \in \tau_1$ of participant p_1 (only 20% best realizations)

- Max

```
_I: Fp2 0.61
MI: CP5 0.69
RI: Pz 0.81
_M: T8 0.67
RIM: CP6 0.65
_IR: Fp2 0.73
```

- Mean

_I	0.51
MI	0.53
RI	0.69
_M	0.52
RIM	0.52
_IR	0.48



* 0.4 >= accuracy >= 0.6