Õie Renata Siimon

NEURO-NLP:
Project idea: discretizing EEG signals or spikes and then forming words and sentences out of them in order to use them as input to an NLP classification model.

3 OPTIONS FOR DISCRETIZING:

a) If using spiking data (seems more feasible): discretize small time intervals into "words" by assigning each electrode/channel a different letter and forming words from letters corresponding to channels that were active. Being active or not can be determined by either defining some threshold (a min number of spikes where it is considered active), or if time interval is very short, then by determining whether there were any spikes in that interval or not.

When choosing the length of the short time intervals, on one hand, total length of time series we want to use, and on the other hand, some reasonable max number of words in sentence needs to be taken into account. If we fix sentence length to be 250 words (which is feasible for NLP models), and total time we consider is 2.5 seconds, then we could use 10ms splits. That is, each word would represent brain activity in one 10 ms interval.

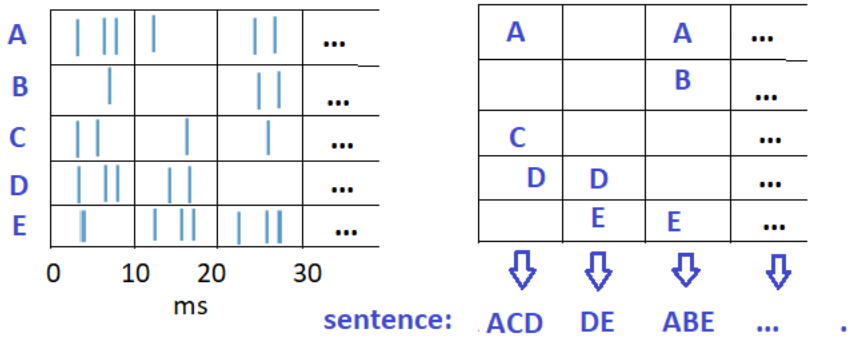

Figure 1. An example of forming words and sentences if we set threshold to 2 spikes. (If there are 2 or more spikes, we use the letter corresponding to the sensor, and if not, we leave this letter out).

An important constraint here is the maximum "vocabulary" size (max number of words that can be formed), which in this case would be $\mathbf{2^n}$, where n is the number of electrodes used. (Letters in a word are always alphabetically ordered, and each letter can either appear or not.) This means that only a limited number of electrodes can be used as input. The maximum feasible number of electrodes would be 12-15 (which would correspond to 4096 or 32768 words, respectively). In NLP models, a vocabulary size of 20k-30k would be ok (in few cases also 50k have been used), but model would likely run slower with larger vocabulary size, would need more data and would be more likely to encounter words in test set that it has not seen in train set (having a few of such words is ok, but having many affects performance).

The NLP models that can be used are word2vec[1] plus any classifier on top of it, or BERT[2]. The output of word2vec are word vectors. In this case, each vector would represent a state in brain activity (which channels were active or not). Those vectors can then be used as input for some classifier (KNN, MLP, etc.). I believe using word2vec seems preferrable for project, since from my experience setting up and running Bert is a time-consuming and difficult process, especially if one does it for the first time. Also, training a Bert model on personal computer is not feasible (should be run in HPC). Also, word2vec would need smaller dataset. For Bert, the dataset size should be at least 5 million, but preferably larger. I don't think it would give any useful results when using dataset size of less than 1 million.

b) if using spiking or EEG data: somehow identify relevant brain areas (max 15 in total) that were active in each short time interval chosen (e.g. 10 ms), by comparing the signals from different sensors. Then proceed in same way as described in point a).

c) If using EEG data (seems less feasible): Split signals into short time intervals and cluster them based on either i) the voltage values themselves (possibly after some smoothing), or ii) statistical measures describing them (mean or median, min, max, standard deviation, etc.). Then designate each cluster with a "letter", and form words from letters corresponding to each of the short time intervals. Then concatenate words into a sentence. The difference from point a) is that in a), we only look if each sensor was active or not, but here, we also take into account which type of pattern of activity each sensor had.

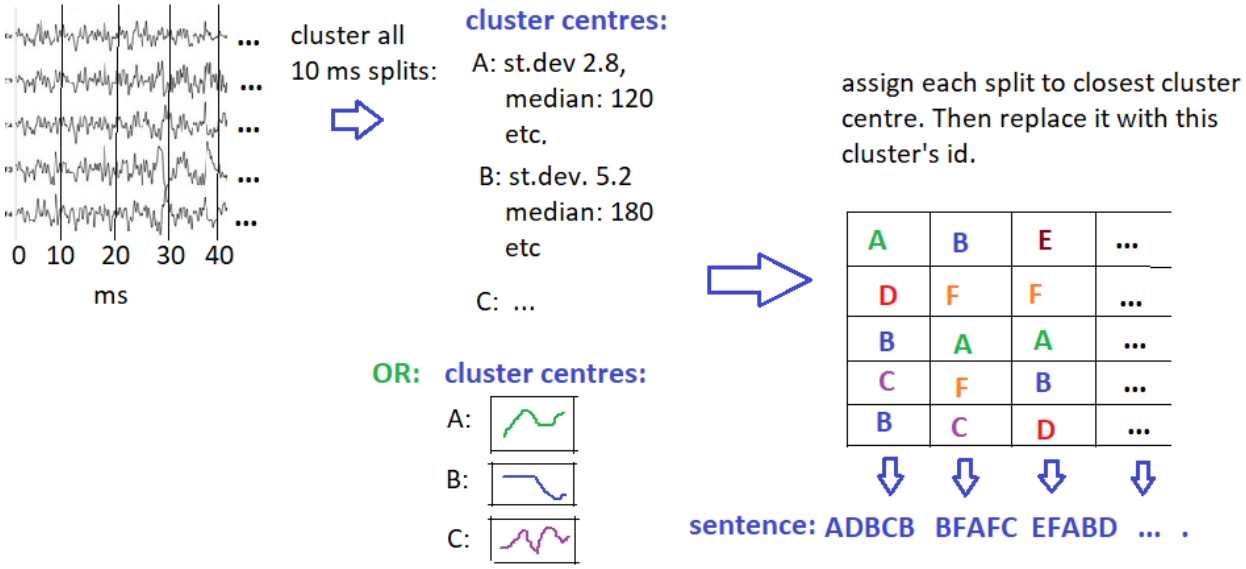

Figure 2. Forming sentences in case of option b).

The problem with this approach is combinatorics, since the max number of words in this case would be $\mathbf{c^n}$, where c is number of clusters and n is the number of sensors. (All words are of the same length, which equals to number of sensors used (n), and each letter in a word can take c values.) Then, for example if we use 30 clusters and 3 sensors, then max number of words would be 30^3=27000. However, using only 3 channels seems too few. But going beyond 3 channels seems unfeasible, because it's an exponent.

---

[1] Gensim word2vec library, https://radimrehurek.com/gensim/models/word2vec.html
[2] Huggingface Transformers library, https://huggingface.co/transformers/model_doc/bert.html